Open Source Used In DNAC 1.3.3 DNAC 1.3.1.0 NDP

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1112656775

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this form.

In your requests please include the following reference number 78EE117C99-1112656775

Contents

1.1 jackson-core-asl 1.9.13
  1.1.1 Available under license
1.2 javax-inject 2.2.0
  1.2.1 Available under license
1.3 paranamer-core 2.7
  1.3.1 Available under license
1.4 githubcomgolangprotobuf 1.0
  1.4.1 Available under license
1.5 commons-collections 3.2.2
  1.5.1 Available under license
1.6 body-parser 1.15.0
  1.6.1 Available under license
1.7 xz 1.5
  1.7.1 Available under license
1.8 express 4.13.4
  1.8.1 Available under license
1.9 reflections 0.9.10
  1.9.1 Available under license
1.10 snappy-java 1.1.2.4
  1.10.1 Available under license
1.11 google-http-client 1.20.0
  1.11.1 Available under license
1.12 jsplumb 2.2.0
  1.12.1 Available under license
1.13 redux-thunk 2.1.0
  1.13.1 Available under license

1.14 antlr 4.5.3
1.14.1 Available under license
1.15 cookie 0.3.1
1.15.1 Available under license
1.16 xerces-j 2.9.1
1.16.1 Available under license
1.17 request 2.81.0
1.17.1 Available under license
1.18 moment-duration-format 1.3.0
1.18.1 Available under license
1.19 asm 2.2.0
1.19.1 Available under license
1.20 xmlschema 2.0.3
1.20.1 Available under license
1.21 bouncy-castle 1.55
1.21.1 Available under license
1.22 commons-pool 2.4.2
1.22.1 Available under license
1.23 commons-collections 3.2.2
1.23.1 Available under license
1.24 commons-logging 1.2
1.24.1 Available under license
1.25 google-protobuf 3.2.0
1.25.1 Available under license
1.26 mime-pull 1.9.3
1.26.1 Available under license
1.27 resourcelocator 1.0.1
1.27.1 Available under license
1.28 spring-tx 3.0.5
1.28.1 Available under license
1.29 expression 3.0.5
1.29.1 Available under license
1.30 spring-asm 3.0.5
1.31 spring-expression 3.0.5
1.31.1 Available under license
1.32 spring-framework 3.0.5
1.32.1 Available under license
1.33 jcip-annotation 1.0-1
1.33.1 Available under license

1.34 json-java 20140107
1.34.1 Available under license
1.35 swagger 1.5.0
1.36 react-router-dom 4.1.1
1.36.1 Available under license
1.37 redux 3.4.1
1.37.1 Available under license
1.38 redux-act 1.3.2
1.38.1 Available under license
1.39 highcharts-no-data-to-display 0.1.2
1.40 rc-table 5.2.2
1.40.1 Available under license
1.41 beam 0.6.0
1.41.1 Available under license
1.42 fst 2.52
1.42.1 Available under license
1.43 tinkerpop-gremlin 3.2
1.43.1 Available under license
1.44 testify 1.1.3
1.44.1 Available under license
1.45 moment 2.19.3
1.45.1 Available under license
1.46 react-redux 5.0.6
1.46.1 Available under license
1.47 commons-compress 1.4.1
1.47.1 Available under license
1.48 kafka-node 1.6.2
1.48.1 Available under license
1.49 antlr 3.5.2
1.49.1 Available under license
1.50 backport-util-concurrent 3.1
1.50.1 Available under license
1.51 commons-digester 1.8.1
1.51.1 Available under license
1.52 commons-io 1.4
1.52.1 Available under license
1.53 hibernate 3.6.3
1.54 javax-websocket-api 1.1
1.54.1 Available under license

1.55 joda-time 2.2

1.55.1 Available under license

1.56 junit 4.10

1.56.1 Available under license

1.57 apache-log4j 2.8.2

1.57.1 Available under license

1.58 logback-core 1.1.7

1.59 metrics-core 2.2.0

1.59.1 Available under license

1.60 spring-aop 3.0.5

1.60.1 Available under license

1.61 spring-beans 3.0.5

1.61.1 Available under license

1.62 spring-context 3.0.5

1.62.1 Available under license

1.63 stax-api 1.0.1

1.63.1 Available under license

1.64 thrift 0.9.2

1.64.1 Available under license

1.65 zlib 1.2.1

1.66 prop-types 15.6.0

1.66.1 Available under license

1.67 commons-codec 1.10

1.67.1 Available under license

1.68 commons-configuration 1.10

1.68.1 Available under license

1.69 jul-to-slf4j 1.7.5

1.69.1 Available under license

1.70 rhino 1.7.7.1

1.70.1 Available under license

1.71 snake-yaml 1.12

1.72 xpp 1.1.4c

1.72.1 Available under license

1.73 avro 1.8.2

1.73.1 Available under license

1.74 httpasyncclient 4.1.2

1.74.1 Available under license

1.75 javassist 3.18.2

1.75.1 Available under license

1.76 tiger-types 1.4
1.76.1 Available under license
1.77 bunyan 1.8.12
1.77.1 Available under license
1.78 commons-codec 1.4
1.78.1 Available under license
1.79 httpcomponents-client 4.5.3
1.79.1 Available under license
1.80 httpcomponents-core 4.4.6
1.80.1 Available under license
1.81 javax-ws-rs-api 2.0.1
1.81.1 Available under license
1.82 reflections 0.9.10
1.83 swagger-jaxrs 1.5.0
1.83.1 Available under license
1.84 swagger-jersey2-jaxrs 1.5.0
1.85 antlr 3.1.3
1.85.1 Available under license
1.86 avro-guava-dependencies 1.8.2
1.86.1 Available under license
1.87 aop-alliance 1.0.0
1.88 grizzly-framework 2.3.25
1.88.1 Available under license
1.89 grizzly-http 2.3.25
1.89.1 Available under license
1.90 grizzly-http-server 2.3.25
1.90.1 Available under license
1.91 jackson-jaxrs-base 2.7.9
1.91.1 Available under license
1.92 javax-el-api 2.2.4
1.92.1 Available under license
1.93 spring-expression 4.3.5
1.93.1 Available under license
1.94 bson4jackson 2.7.0
1.94.1 Available under license
1.95 spring-framework 4.3.5
1.95.1 Available under license
1.96 okio 1.13.0
1.96.1 Available under license

1.97 cxf 2.7.3
1.97.1 Available under license
1.98 cxf-rt 2.7.3
1.98.1 Available under license
1.99 aspect-j 1.8.13
1.99.1 Available under license
1.100 byte-buddy 1.7.10
1.100.1 Available under license
1.101 javatuples 1.2
1.101.1 Available under license
1.102 javax-el 2.2.4
1.102.1 Available under license
1.103 jaxb-api 2.2.11
1.103.1 Available under license
1.104 jedis 2.8.1
1.104.1 Available under license
1.105 joda-time 2.7
1.105.1 Available under license
1.106 json-smart 1.3.1
1.106.1 Available under license
1.107 jtoml 1.0.0
1.108 log4j-api 2.8
1.108.1 Available under license
1.109 log4j-slf4j-impl 2.8
1.109.1 Available under license
1.110 httpcomponents-client 4.5.4
1.110.1 Available under license
1.111 metrics-health-checks 3.1.0
1.112 metrics-jetty 3.1.0
1.113 metrics-jvm 3.1.0
1.114 mongodb-driver 3.0.4
1.114.1 Available under license
1.115 mongodb-driver-core 3.0.4
1.115.1 Available under license
1.116 stax 3.1.4
1.116.1 Available under license
1.117 tyrus-client 1.13
1.117.1 Available under license
1.118 tyrus-container-grizzly-client 1.13

1.118.1 Available under license
1.119 tyrus-core 1.13
1.119.1 Available under license
1.120 tyrus-spi 1.13
1.120.1 Available under license
1.121 st4 4.0.8
1.121.1 Available under license
1.122 netty 3.10.5
1.122.1 Available under license
1.123 abego-treelayout-core 1.0.1
1.123.1 Available under license
1.124 netty 4.1.11
1.124.1 Available under license
1.125 bcpkix-jdk15on 1.55
1.126 amqp-client 3.6.2
1.127 commons-validator 1.5.1
1.127.1 Available under license
1.128 jackson-jaxrs-base 2.7.4
1.128.1 Available under license
1.129 netty 4.1.16
1.129.1 Available under license
1.130 minlog 1.2
1.131 antlr-runtime 4.5.3
1.131.1 Available under license
1.132 cassandra-driver-mapping 3.0.2
1.132.1 Available under license
1.133 google-api-client 1.22.0
1.133.1 Available under license
1.134 googleapiclientjackson 1.20.0
1.134.1 Available under license
1.135 googleapiclientjava 1.20.0
1.135.1 Available under license
1.136 google-http-client 1.22.0
1.136.1 Available under license
1.137 google-oauth-client 1.22.0
1.137.1 Available under license
1.138 groovy-console 2.4.6
1.138.1 Available under license
1.139 groovy-groovysh 2.4.6

1.139.1 Available under license
1.140 groovy-json 2.4.6
1.140.1 Available under license
1.141 groovyjsr 2.4.6
1.141.1 Available under license
1.142 groovy-swing 2.4.6
1.142.1 Available under license
1.143 groovy-templates 2.4.6
1.143.1 Available under license
1.144 groovy-xml 2.4.6
1.144.1 Available under license
1.145 grpc-okhttp 0.12.0
1.145.1 Available under license
1.146 grpc-protobuf 0.12.0
1.146.1 Available under license
1.147 grpc-protobuf-nano 0.12.0
1.147.1 Available under license
1.148 ivy 2.3.0
1.148.1 Available under license
1.149 jackson 2.7.2
1.149.1 Available under license
1.150 javax-json 1.0.4
1.150.1 Available under license
1.151 javax-json-api 1.0
1.151.1 Available under license
1.152 jbcrypt 0.4
1.152.1 Available under license
1.153 jcabi-log 0.14
1.153.1 Available under license
1.154 jcabi-manifests 1.1
1.154.1 Available under license
1.155 jline 2.12
1.155.1 Available under license
1.156 json-java 20160810
1.156.1 Available under license
1.157 minlog 1.3.0
1.158 okhttp 2.5.0
1.158.1 Available under license
1.159 everit-json-schema 1.4.1

1.159.1 Available under license
1.160 protobuf-javanano 3.0.0-alpha-4
1.160.1 Available under license
1.161 reflectasm 1.10.1
1.161.1 Available under license
1.162 util 1.4.5
1.162.1 Available under license
1.163 config 1.3.0
1.164 asm-all-repackaged 2.2.0
1.164.1 Available under license
1.165 cglib 2.2.0
1.165.1 Available under license
1.166 hk2-api 2.2.0
1.166.1 Available under license
1.167 hk2-locator 2.2.0
1.167.1 Available under license
1.168 hk2-utils 2.2.0
1.168.1 Available under license
1.169 xstream 1.3.1
1.169.1 Available under license
1.170 httpcore-nio 4.4.6
1.170.1 Available under license
1.171 react-bootstrap 0.32.1
1.171.1 Available under license
1.172 aspectj-tools 1.8.13
1.173 gson 2.8.2
1.173.1 Available under license
1.174 commons-beanutils 1.9.2
1.174.1 Available under license
1.175 protobuf-java 3.2.0
1.175.1 Available under license
1.176 protobuf-java-util 3.2.0
1.176.1 Available under license
1.177 request-local 1.0.5
1.177.1 Available under license
1.178 react 16.2.0
1.178.1 Available under license
1.179 react-dom 16.2.0
1.179.1 Available under license

1.180 metrics 3.1.0
1.180.1 Available under license
1.181 metrics 3.1.5
1.181.1 Available under license
1.182 okhttp 3.9.1
1.182.1 Available under license
1.183 jackson-annotations 2.9.0
1.183.1 Available under license
1.184 log4j-api 2.9.1
1.184.1 Available under license
1.185 httpasyncclient 4.1.3
1.185.1 Available under license
1.186 json-simple 1.1.1
1.186.1 Available under license
1.187 scala 2.11.12
1.187.1 Available under license
1.188 zookeeper 3.4.10
1.188.1 Available under license
1.189 cxf-rt-transports-http 3.2.6
1.189.1 Available under license
1.190 cxf-rt-rs-client 3.2.6
1.190.1 Available under license
1.191 annotations 2.0.1
1.191.1 Available under license
1.192 compiler 0.9.3
1.193 jackson-dataformat-cbor 2.8.10
1.194 jackson-dataformat-smile 2.8.10
1.194.1 Available under license
1.195 jackson-dataformat-yaml 2.8.10
1.195.1 Available under license
1.196 jna 4.5.1
1.197 bzip2 1.0.6
1.197.1 Available under license
1.198 core-module 1.1.7
1.199 typesafe-config 1.3.0
1.200 commons-csv 1.4
1.200.1 Available under license
1.201 animal-sniffer-annotation 1.14
1.202 hdrhistogram 2.1.9

1.203 joda-time 2.9.9
1.203.1 Available under license
1.204 zlib 1.2.11
1.204.1 Available under license
1.205 javax-inject 1
1.206 jackson 2.8.10
1.206.1 Available under license
1.207 checker-compat-qual 2.0.0
1.208 commons-codec 1.6
1.208.1 Available under license
1.209 commons-io 2.5
1.209.1 Available under license
1.210 guava 25.0
1.211 lz4-java 1.4
1.212 osgi-core 6.0.0
1.212.1 Available under license
1.213 slf4j 1.7.25
1.214 snappy-java 1.1.4
1.215 metrics 3.1.2
1.216 snappy 6.1.1
1.216.1 Available under license
1.217 lucene 7.3.1
1.217.1 Available under license
1.218 lucene-analyzers-common 7.3.1
1.218.1 Available under license
1.219 lucene-backward-codecs 7.3.1
1.219.1 Available under license
1.220 lucene-grouping 7.3.1
1.220.1 Available under license
1.221 lucene-highlighter 7.3.1
1.221.1 Available under license
1.222 lucene-join 7.3.1
1.222.1 Available under license
1.223 lucene-memory 7.3.1
1.223.1 Available under license
1.224 lucene-misc 7.3.1
1.224.1 Available under license
1.225 lucene-queries 7.3.1
1.225.1 Available under license

1.226 lucene-queryparser 7.3.1
1.226.1 Available under license
1.227 lucene-sandbox 7.3.1
1.227.1 Available under license
1.228 lucene-spatial 7.3.1
1.228.1 Available under license
1.229 lucene-spatial-extras 7.3.1
1.229.1 Available under license
1.230 lucene-spatial3d 7.3.1
1.230.1 Available under license
1.231 lucene-suggest 7.3.1
1.231.1 Available under license
1.232 netty 4.1.16
1.233 netty-buffer 4.1.16
1.234 netty-codec 4.1.16
1.235 netty-codec-http 4.1.16
1.236 netty-handler 4.1.16
1.237 netty-resolver 4.1.16
1.238 netty-transport 4.1.16
1.239 t-digest 3.2
1.240 commons-collections 4.1
1.240.1 Available under license
1.241 satori-go-uuid 1.2.0
1.241.1 Available under license
1.242 httpcomponents-core 4.4.5
1.242.1 Available under license
1.243 httpcore-nio 4.4.5
1.243.1 Available under license
1.244 jopt-simple 5.0.2
1.245 spatial4j 0.7
1.246 google-go-cmp 0.2.0
1.246.1 Available under license
1.247 go-profile 1.2.1
1.247.1 Available under license
1.248 annotations 3.0.1
1.249 netty 4.0.27
1.250 golang-snappy 2e65f85
1.250.1 Available under license
1.251 groovy 2.4.12

1.252 reactive-streams 1.0.0
1.252.1 Available under license
1.253 swagger-annotations 1.5.16
1.253.1 Available under license
1.254 olivere-elastic 6.1.27
1.254.1 Available under license
1.255 cglib 3.2.0
1.255.1 Available under license
1.256 okhttp 3.9.0
1.257 go-spew v1.1.1
1.257.1 Available under license
1.258 go-resiliency v1.1.0
1.258.1 Available under license
1.259 go-xerial-snappy 776d571
1.259.1 Available under license
1.260 eapache-queue v1.1.0
1.260.1 Available under license
1.261 pierrec-lz4 v2.0.7
1.261.1 Available under license
1.262 cassandra-driver-core 3.0.2
1.263 classmate 0.8.0
1.264 hibernate-validator 5.0.0
1.265 jackson-dataformat-xml 2.7.9
1.265.1 Available under license
1.266 jackson-dataformat-yaml 2.4.2
1.266.1 Available under license
1.267 jackson-datatype-joda 2.4.2
1.267.1 Available under license
1.268 jackson-jaxrs-xml-provider 2.7.9
1.268.1 Available under license
1.269 java-jwt 2.1.0
1.270 jboss-logging 3.1.1
1.270.1 Available under license
1.271 jersey-bean-validation 2.5.1
1.272 jersey-client 2.5.1
1.273 jersey-common 2.5.1
1.274 jersey-container-servlet 2.5.1
1.275 jersey-container-servlet-core 2.5.1
1.276 jersey-media-multipart 2.5.1

1.277 jersey-server 2.5.1
1.278 metrics-json 3.1.0
1.279 minio 4.0.0
1.280 netty 4.1.27
1.281 spring-aspects 3.0.5
1.282 swagger-annotations 1.5.0
1.282.1 Available under license
1.283 antlr 4.7.1
1.284 antlr-runtime 4.7.1
1.285 cglib 3.2.6
1.285.1 Available under license
1.286 chill-java 0.7.4
1.287 classindex 3.4
1.288 commons-math 3.5
1.288.1 Available under license
1.289 disruptor 3.3.2
1.290 druid 1.0.15
1.291 elasticsearch-cli 6.3.1
1.291.1 Available under license
1.292 elasticsearch-core 6.3.1
1.292.1 Available under license
1.293 elasticsearch-rest-client 6.3.1
1.293.1 Available under license
1.294 elasticsearch-x-content 6.3.1
1.294.1 Available under license
1.295 gbench 0.4.3
1.296 google-auth-library-credentials 0.4.0
1.297 grpc-auth 0.12.0
1.298 grpc-core 0.12.0
1.299 grpc-netty 0.12.0
1.300 grpc-stub 0.12.0
1.301 gson-fire 1.5.1
1.302 guice 4.2.0
1.302.1 Available under license
1.303 hpack 0.10.1
1.304 ical4j 1.0.2
1.305 icu4j 58.2
1.305.1 Available under license
1.306 jcl-over-slf4j 1.7.19

1.307 jopt-simple 5.0.4
1.308 json-schema-validator 0.1.16
1.309 jsoup 1.9.2
1.310 jython 2.5.2
1.311 kafka-dropwizard-reporter 1.1.1
1.312 kryo 2.24.0
1.313 lang-mustache-client 6.3.1
1.313.1 Available under license
1.314 mvel2 2.2.1
1.315 natty 0.13
1.316 netty 4.0.51
1.317 reindex-client 6.3.1
1.317.1 Available under license
1.318 remark 1.0.0
1.319 rhino 1.7.7.2
1.320 rocksdbjni 5.6.1
1.321 scala 2.11.11
1.322 siddhi-annotations 4.1.33
1.322.1 Available under license
1.323 siddhi-core 4.1.33
1.323.1 Available under license
1.324 siddhi-execution-regex 4.0.10
1.324.1 Available under license
1.325 siddhi-execution-string 4.0.11
1.325.1 Available under license
1.326 siddhi-query-api 4.1.33
1.326.1 Available under license
1.327 siddhi-query-compiler 4.1.33
1.327.1 Available under license
1.328 slf4j 1.7.19
1.329 slf4j-ext 1.7.25
1.330 transport 6.3.1
1.330.1 Available under license
1.331 transport-netty4-client 6.3.1
1.331.1 Available under license
1.332 zjsonpatch 0.3.6
1.332.1 Available under license
1.333 commons-http-client 3.1
1.334 d3 4.12.0

1.334.1 Available under license
1.335 react-jsonschema-form 1.0.0
1.335.1 Available under license
1.336 react-dates 15.2.0
1.336.1 Available under license
1.337 d3sankeydiagram 0.7.1
1.337.1 Available under license
1.338 jackson 2.9.8
1.338.1 Available under license
1.339 cxf 3.2.6
1.339.1 Available under license
1.340 google-http-client 1.27.0
1.341 groovy-all 2.5.6
1.342 groovy-templates 2.5.6
1.343 groovy-console 2.5.6
1.344 groovy-xml 2.5.6
1.345 groovy-groovysh 2.5.6
1.346 bson 3.0.4
1.347 metrics-servlets 3.1.0
1.348 jackson-annotations 2.9.8
1.348.1 Available under license
1.349 sarama-shopify 1.19.0
1.349.1 Available under license
1.350 aop-alliance 1.0
1.351 xpp3-min 1.1.4c
1.352 jquery 3.4.0
1.352.1 Available under license
1.353 cxf-rt-frontend-jaxrs 3.2.6
1.353.1 Available under license
1.354 paesslerag-jsonpath 0.1.1
1.354.1 Available under license
1.355 jackson 2.7.9
1.356 jackson-core-asl 1.9.13
1.357 woodstox-core-asl 4.4.1
1.358 snappy-java 1.1.7.1
1.359 jcip-annotation 1.0
1.360 commons-cli 1.4
1.360.1 Available under license
1.361 zkclient 0.10

1.362 c3p0 0.9.1.1

1.362.1 Available under license

1.363 jcommander 1.48

1.364 smackjava 4.1.9

1.365 javassist 3.21.0

1.366 httpcomponents-client 4.5.2

1.366.1 Available under license

1.367 commons-io 2.4

1.367.1 Available under license

1.368 json-path 2.2.0

1.368.1 Available under license

1.369 jackson-annotations 2.8.0

1.369.1 Available under license

1.370 commons-io 1.3.2

1.370.1 Available under license

1.371 scala-reflect 2.11.12

1.372 httpcomponents-client 4.2.5

1.372.1 Available under license

1.373 httpcomponents-core 4.2.4

1.373.1 Available under license

1.374 smack-tcp 4.1.9

1.375 smack-resolver-javax 4.1.9

1.376 smack-im 4.1.9

1.377 smack-sasl-javax 4.1.9

1.378 smack-extensions 4.1.9

1.379 sprintf-js 1.0.3

1.379.1 Available under license

1.380 javax-annotation-api 1.2

1.380.1 Available under license

1.381 grpc-grpc-common 1.0

1.381.1 Available under license

1.382 jackson-databind 2.9.9

1.382.1 Available under license

1.383 lodash 4.17.11

1.383.1 Available under license

1.384 rabbitmq-java-client 3.6.2

1.384.1 Available under license

1.385 jsr305 1.3.9

1.385.1 Available under license

1.386 commons-lang3 3.4
1.386.1 Available under license
1.387 socket.io-client 2.0.1
1.387.1 Available under license
1.388 goavro 2.7.0
1.388.1 Available under license
1.389 tsenart-tb 19f4c3d
1.389.1 Available under license
1.390 socket.io 2.0.1
1.390.1 Available under license
1.391 commons-lang3 3.6
1.391.1 Available under license
1.392 bean-validation-api 1.1.0
1.392.1 Available under license
1.393 flink-metrics-core 1.4.0
1.393.1 Available under license
1.394 jaeger-core 0.23.0
1.394.1 Available under license
1.395 jsr305 3.0.1
1.395.1 Available under license
1.396 google-http-client-xml 1.20.0
1.396.1 Available under license
1.397 flink-connector-kafka 2.11-1.4.0
1.397.1 Available under license
1.398 scala-logging_2.12 3.7.2
1.398.1 Available under license
1.399 commons-lang3 3.5
1.399.1 Available under license
1.400 apache-log4j 2.8
1.400.1 Available under license
1.401 jaeger-thrift 0.23.0
1.401.1 Available under license
1.402 commons-lang3 2.6
1.402.1 Available under license
1.403 commons-lang3 3.2.1
1.403.1 Available under license
1.404 commons-lang3 3.3.1
1.404.1 Available under license
1.405 groovy-sql 2.4.6

1.405.1 Available under license
1.406 slf4j-log4j 1.7.19
1.406.1 Available under license
1.407 flink-core 1.4.0
1.407.1 Available under license
1.408 commons-codec 1.9
1.408.1 Available under license
1.409 error_prone_annotations 2.1.3
1.409.1 Available under license
1.410 j2objc-annotations 1.1
1.410.1 Available under license
1.411 hppc 0.7.1
1.411.1 Available under license
1.412 woodstox-core 5.1.0
1.412.1 Available under license
1.413 commons-logging 1.1.1
1.413.1 Available under license
1.414 commons-cli 1.3.1
1.414.1 Available under license
1.415 jackson-jaxrs 2.7.9
1.415.1 Available under license
1.416 servlet-api 3.1.0
1.416.1 Available under license
1.417 google-http-client-jackson2 1.22.0
1.417.1 Available under license
1.418 google-http-client-jackson2 1.20.0
1.418.1 Available under license
1.419 jackson-xc 2.7.9
1.419.1 Available under license
1.420 elasticsearch-secure-sm 6.3.1
1.420.1 Available under license
1.421 jackson-mapper-asl 1.9.13-ASL License
1.421.1 Available under license

1.1 jackson-core-asl 1.9.13

1.1.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii)
beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the
Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed
under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can
use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of

it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or
for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally,
software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public

License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case,
there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library,
whereas the latter must
be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").

Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim
or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute
verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so

that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion
or
derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However,
linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data

structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a
derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution
displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking
with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any

attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated
so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third
parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to
apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder
who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the
Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE
RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the
public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public

License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

```
/* Jackson JSON-processor.
*
* Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code and binary code bundles.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

=== Licensing ===

Jackson can be used for any purpose, but to (re)distribute it, distributors
(such as libraries and frameworks that use Jackson) will need to
choose which License they want to apply to distribution, and to use appropriate
jars that enclose license documentation.
No work needed beyond choosing the appropriate jar(s).

Currently two Open Source licenses are available for use:

* Apache License 2.0 (AL 2.0)
* Lesser/Library General Public License (LGPL 2.1)

These licenses have proven adequate to cover all current use cases.

1.2 javax-inject 2.2.0

1.2.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License
means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent
Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free,
non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii)
the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor
(or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in
Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter
any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty,
support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may
create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this

License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim
against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT,
SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. § 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent
with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and
venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors
to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom
to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights
or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally,
any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.
(Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring
a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there
is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent
of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge
no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed
(in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept
this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason
(not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents
or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in

reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission.
For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED
TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey
 the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic
 and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James
 Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the
GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

```
/*
* Copyright (C) 2009 The JSR-330 Expert Group
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

1.3 paranamer-core 2.7

1.3.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/***
*
* Copyright (c) 2009 Paul Hammant
* All rights reserved.
```

```
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/AnnotationParanamer.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2007 Paul Hammant
* Copyright 2013 Samuel Halliday
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
```

*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/JavadocParanamer.java
No license file was found, but licenses were detected in source scan.

/***
*
* Copyright (c) 2007 Paul Hammant
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF

* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/AdaptiveParanamer.java
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/DefaultParanamer.java
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/CachingParanamer.java
*
 /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/ParameterNamesNotFoundException.java
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/NullParanamer.java
No license file was found, but licenses were detected in source scan.

/**
*
* Copyright (c) 2013 Stefan Fleiter
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/PositionalParanamer.java
No license file was found, but licenses were detected in source scan.

```
/***
*
* Portions Copyright (c) 2007 Paul Hammant
* Portions copyright (c) 2000-2007 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
 IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/BytecodeReadingParanamer.java
No license file was found, but licenses were detected in source scan.

```
/***
*
* Copyright (c) 2007 Paul Hammant
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
```

Found in path(s):
* /opt/cola/permits/105038887_1695284973.027553/0/fsrc-paranamer-2-7-sources-jar/com/thoughtworks/paranamer/Paranamer.java

1.4 githubcomgolangprotobuf 1.0

1.4.1 Available under license :

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.5 commons-collections 3.2.2

1.5.1 Available under license :

Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.6 body-parser 1.15.0

1.6.1 Available under license :

(The MIT License)

1.7 xz 1.5

1.7.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* CorruptedInputException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/CorruptedInputException.java

No license file was found, but licenses were detected in source scan.

/*
* LZDecoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/LZDecoder.java
No license file was found, but licenses were detected in source scan.

/*
* SimpleInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SimpleInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* LZMA2Options
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMA2Options.java
No license file was found, but licenses were detected in source scan.

/*
* Optimum
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*

```
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/Optimum.java

No license file was found, but licenses were detected in source scan.

```
/*
* LZMACoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/LZMACoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* BCJEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/BCJEncoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* IndexDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/index/IndexDecoder.java

No license file was found, but licenses were detected in source scan.

/*
* LZMA2Decoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMA2Decoder.java
No license file was found, but licenses were detected in source scan.

/*
* DeltaEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/DeltaEncoder.java
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/delta/DeltaEncoder.java
No license file was found, but licenses were detected in source scan.

/*
* LZMAEncoderNormal
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/LZMAEncoderNormal.java
No license file was found, but licenses were detected in source scan.

/*
* Check
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.

```
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/check/Check.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for little endian ARM-Thumb instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/ARMThumb.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for x86 instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/X86.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/XZInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* DeltaOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/DeltaOptions.java
No license file was found, but licenses were detected in source scan.

/*
* XZOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/XZOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* XZIOException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/XZIOException.java
No license file was found, but licenses were detected in source scan.

/*
* RangeEncoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/rangecoder/RangeEncoder.java
No license file was found, but licenses were detected in source scan.

/*
* UnsupportedOptionsException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/UnsupportedOptionsException.java
No license file was found, but licenses were detected in source scan.

/*
* BCJ filter for big endian PowerPC instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/PowerPC.java
No license file was found, but licenses were detected in source scan.

/*
* LZMADecoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/LZMADecoder.java
No license file was found, but licenses were detected in source scan.

/*
* FilterDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/FilterDecoder.java
No license file was found, but licenses were detected in source scan.

/*
* CRC64
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/check/CRC64.java
No license file was found, but licenses were detected in source scan.

/*
* BCJ filter for little endian ARM instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/ARM.java
No license file was found, but licenses were detected in source scan.

/*
* SeekableFileInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SeekableFileInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* IA64Options
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/IA64Options.java
No license file was found, but licenses were detected in source scan.

/*
* DeltaOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/DeltaOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* DeltaDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/delta/DeltaDecoder.java
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/DeltaDecoder.java
No license file was found, but licenses were detected in source scan.

/*
* FilterCoder

*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/FilterCoder.java
No license file was found, but licenses were detected in source scan.

/*
* SimpleOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SimpleOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* BlockInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/BlockInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* LZMA2OutputStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMA2OutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* StreamFlags
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/common/StreamFlags.java
No license file was found, but licenses were detected in source scan.

/*
* RangeDecoderFromBuffer
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/rangecoder/RangeDecoderFromBuffer.java
No license file was found, but licenses were detected in source scan.

/*
* SingleXZInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SingleXZInputStream.java
No license file was found, but licenses were detected in source scan.

/*

```
* FilterOptions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/FilterOptions.java

No license file was found, but licenses were detected in source scan.

```
/*
* ARMOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/ARMOptions.java

No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for SPARC instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/SPARC.java

No license file was found, but licenses were detected in source scan.

```
/*
* IndexEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/index/IndexEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* SeekableInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SeekableInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZFormatException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/XZFormatException.java
No license file was found, but licenses were detected in source scan.

```
/*
* SeekableXZInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SeekableXZInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BlockOutputStream
*
```

* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/BlockOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* IndexHash
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/index/IndexHash.java
No license file was found, but licenses were detected in source scan.

/*
* BCJCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/BCJCoder.java
No license file was found, but licenses were detected in source scan.

/*
* EncoderUtil
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-
jar/org/tukaani/xz/common/EncoderUtil.java

No license file was found, but licenses were detected in source scan.

```
/*
* CountingInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/CountingInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* 2-, 3-, and 4-byte hashing
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/Hash234.java
No license file was found, but licenses were detected in source scan.

```
/**
* XZ data compression support.
*
* <h4>Introduction</h4>
* <p>
* This aims to be a complete implementation of XZ data compression
* in pure Java. Features:
* <ul>
* <li>Full support for the .xz file format specification version 1.0.4</li>
* <li>Single-threaded streamed compression and decompression</li>
* <li>Single-threaded decompression with limited random access support</li>
* <li>Raw streams (no .xz headers) for advanced users, including LZMA2
*     with preset dictionary</li>
* </ul>
* <p>
* Threading is planned but it is unknown when it will be implemented.
* <p>
* For the latest source code, see the
* <a href="http://tukaani.org/xz/java.html">home page of XZ for Java</a>.
*
* <h4>Getting started</h4>
* <p>
* Start by reading the documentation of {@link org.tukaani.xz.XZOutputStream}
* and {@link org.tukaani.xz.XZInputStream}.
* If you use XZ inside another file format or protocol,
 * see also {@link org.tukaani.xz.SingleXZInputStream}.
*
* <h4>Licensing</h4>
* <p>
* XZ for Java has been put into the public domain, thus you can do
* whatever you want with it. All the files in the package have been
* written by Lasse Collin and/or Igor Pavlov.
* <p>
* This software is provided "as is", without any warranty.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
* SHA256
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/check/SHA256.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZEncoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*         Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/LZEncoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* Util
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/common/Util.java
No license file was found, but licenses were detected in source scan.

```
/*
* DeltaCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/DeltaCoder.java
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/delta/DeltaCoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for little endian ARM instructions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/SimpleFilter.java
No license file was found, but licenses were detected in source scan.

```
/*
* Hash Chain match finder with 2-, 3-, and 4-byte hashing
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
```

* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/HC4.java
No license file was found, but licenses were detected in source scan.

/*
* IndexIndicatorException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/IndexIndicatorException.java
No license file was found, but licenses were detected in source scan.

/*
* FinishableWrapperOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/FinishableWrapperOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* None
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/check/None.java

No license file was found, but licenses were detected in source scan.

```
/*
* RawCoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/RawCoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* LZMA2Coder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMA2Coder.java

No license file was found, but licenses were detected in source scan.

```
/*
* BCJ filter for Itanium (IA-64) instructions
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/simple/IA64.java

No license file was found, but licenses were detected in source scan.

```
/*
* CRC32Hash
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
```

* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/CRC32Hash.java
No license file was found, but licenses were detected in source scan.

/*
* LZMA2Encoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMA2Encoder.java
No license file was found, but licenses were detected in source scan.

/*
* FinishableOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/FinishableOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* DecoderUtil
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/common/DecoderUtil.java

No license file was found, but licenses were detected in source scan.

```
/*
* BCJDecoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/BCJDecoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* IndexRecord
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/index/IndexRecord.java
No license file was found, but licenses were detected in source scan.

```
/*
* CRC32
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/check/CRC32.java
No license file was found, but licenses were detected in source scan.

```
/*
* UncompressedLZMA2OutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
```

*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/UncompressedLZMA2OutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* CountingOutputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/CountingOutputStream.java
No license file was found, but licenses were detected in source scan.

/*
* MemoryLimitException
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/MemoryLimitException.java
No license file was found, but licenses were detected in source scan.

/*
* ARMThumbOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/ARMThumbOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* Binary Tree match finder with 2-, 3-, and 4-byte hashing
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*         Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/BT4.java
No license file was found, but licenses were detected in source scan.

```
/*
* RangeDecoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*         Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/rangecoder/RangeDecoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* Matches
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*         Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lz/Matches.java
No license file was found, but licenses were detected in source scan.

```
/*
* LZMAInputStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*         Igor Pavlov <http://7-zip.org/>
```

*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMAInputStream.java
No license file was found, but licenses were detected in source scan.

/*
* LZMAEncoderFast
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*   Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/LZMAEncoderFast.java
No license file was found, but licenses were detected in source scan.

/*
* BCJOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/BCJOptions.java
No license file was found, but licenses were detected in source scan.

/*
* PowerPCOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/PowerPCOptions.java

No license file was found, but licenses were detected in source scan.

```
/*
* State
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/State.java
No license file was found, but licenses were detected in source scan.

```
/*
* RangeDecoderFromStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/rangecoder/RangeDecoderFromStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* XZ
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/XZ.java
No license file was found, but licenses were detected in source scan.

```
/*
* X86Options
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
```

*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/X86Options.java
No license file was found, but licenses were detected in source scan.

```
/*
* SPARCOptions
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/SPARCOptions.java
No license file was found, but licenses were detected in source scan.

```
/*
* DeltaInputStream
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/DeltaInputStream.java
No license file was found, but licenses were detected in source scan.

```
/*
* BlockInfo
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/index/BlockInfo.java
No license file was found, but licenses were detected in source scan.

```
/*
* RangeCoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/rangecoder/RangeCoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* LZMAEncoder
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/lzma/LZMAEncoder.java

No license file was found, but licenses were detected in source scan.

```
/*
* LZMA2InputStream
*
* Authors: Lasse Collin <lasse.collin@tukaani.org>
*          Igor Pavlov <http://7-zip.org/>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):

* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/LZMA2InputStream.java

No license file was found, but licenses were detected in source scan.

```
/*
* FilterEncoder
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
```

* This file has been put into the public domain.
* You can do whatever you want with this file.
*/

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/FilterEncoder.java
No license file was found, but licenses were detected in source scan.

```
/*
* IndexBase
*
* Author: Lasse Collin <lasse.collin@tukaani.org>
*
* This file has been put into the public domain.
* You can do whatever you want with this file.
*/
```

Found in path(s):
* /opt/cola/permits/127605293_1695293501.8677952/0/xz-1-5-sources-jar/org/tukaani/xz/index/IndexBase.java

1.8 express 4.13.4

1.8.1 Available under license :

(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2013-2014 Roman Shtylman <shtylman+expressjs@gmail.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT
SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.9 reflections 0.9.10

1.9.1 Available under license :

```
            DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
                    Version 2, December 2004

 Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

 Everyone is permitted to copy and distribute verbatim or modified
 copies of this license document, and changing it is allowed as long
 as the name is changed.

            DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. You just DO WHAT THE FUCK YOU WANT TO.
```

1.10 snappy-java 1.1.2.4

1.10.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements.  See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership.  The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License.  You may obtain a copy of the License at
* <p/>
* http://www.apache.org/licenses/LICENSE-2.0
* <p/>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
* <p/>
* Some portions of this file Copyright (c) 2004-2006 Intel Corportation
* and licensed under the BSD license.
 */
```

Found in path(s):

* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/PureJavaCrc32C.java
No license file was found, but licenses were detected in source scan.

```
/*--------------------------------------------------------------------------
*  Copyright 2008 Taro L. Saito
*
*  Licensed under the Apache License, Version 2.0 (the "License");
*  you may not use this file except in compliance with the License.
*  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing, software
*  distributed under the License is distributed on an "AS IS" BASIS,
*  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*  See the License for the specific language governing permissions and
*  limitations under the License.
*--------------------------------------------------------------------------*/
```

Found in path(s):
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/OSInfo.java
No license file was found, but licenses were detected in source scan.

```
/*--------------------------------------------------------------------------
*  Copyright 2011 Taro L. Saito
*
*  Licensed under the Apache License, Version 2.0 (the "License");
*  you may not use this file except in compliance with the License.
*  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing, software
*  distributed under the License is distributed on an "AS IS" BASIS,
*  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*  See the License for the specific language governing permissions and
*  limitations under the License.
*--------------------------------------------------------------------------*/
```

Found in path(s):
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyError.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/Snappy.java
*

/opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyException.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyErrorCode.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyBundleActivator.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyOutputStream.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/package-info.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyLoader.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyCodec.java
* /opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyNative.java
*
/opt/cola/permits/132700017_1695296100.1185179/0/snappy-java-1-1-2-4-sources-jar/org/xerial/snappy/SnappyInputStream.java

1.11 google-http-client 1.20.0

1.11.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

BSD License

Copyright (c) 2000-2006, www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.12 jsplumb 2.2.0

1.12.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**

* jsBezier-0.8

*

*

* a set of Bezier curve functions that deal with Beziers, used by jsPlumb, and perhaps useful for other people. These functions work with Bezier

* curves of arbitrary degree.

*

* - functions are all in the 'jsBezier' namespace.

*

* - all input points should be in the format {x:.., y:..}. all output points are in this format too.

*

* - all input curves should be in the format [{x:.., y:..}, {x:.., y:..}, {x:.., y:..}, {x:.., y:..}]

*

* - 'location' as used as an input here refers to a decimal in the range 0-1 inclusive, which indicates a point some proportion along the length

* of the curve. location as output has the same format and meaning.

*

*

* Function List:

* --------------

*

* distanceFromCurve(point, curve)

*

* Calculates the distance that the given point lies

 from the given Bezier. Note that it is computed relative to the center of the Bezier,

* so if you have stroked the curve with a wide pen you may wish to take that into account! The distance returned is relative to the values

* of the curve and the point - it will most likely be pixels.

*

* gradientAtPoint(curve, location)

*

* Calculates the gradient to the curve at the given location, as a decimal between 0 and 1 inclusive.

*

* gradientAtPointAlongCurveFrom (curve, location)

*

* Calculates the gradient at the point on the given curve that is 'distance' units from location.

*

* nearestPointOnCurve(point, curve)

```
*
* Calculates the nearest point to the given point on the given curve.  The return value of this is a JS object literal,
containing both the
*point's coordinates and also the 'location' of the point (see above), for example:  { point:{x:551,y:150},
location:0.263365 }.
*
* pointOnCurve(curve, location)
*
*  Calculates the coordinates of the point on the given
 Bezier curve at the given location.
*
* pointAlongCurveFrom(curve, location, distance)
*
*  Calculates the coordinates of the point on the given curve that is 'distance' units from location.  'distance' should be
in the same coordinate
* space as that used to construct the Bezier curve.  For an HTML Canvas usage, for example, distance would be a
measure of pixels.
*
* locationAlongCurveFrom(curve, location, distance)
*
*  Calculates the location on the given curve that is 'distance' units from location.  'distance' should be in the same
coordinate
* space as that used to construct the Bezier curve.  For an HTML Canvas usage, for example, distance would be a
measure of pixels.
*
* perpendicularToCurveAt(curve, location, length, distance)
*
*  Calculates the perpendicular to the given curve at the given location.  length is the length of the line you wish for
(it will be centered
* on the point at 'location'). distance is optional, and allows you to specify a point along
 the path from the given location as the center of
* the perpendicular returned.  The return value of this is an array of two points: [ {x:...,y:...}, {x:...,y:...} ].
*
*
*/
/**
* Biltong v0.3
*
* Various geometry functions written as part of jsPlumb and perhaps useful for others.
*
* Copyright (c) 2016 jsPlumb
* https://jsplumbtoolkit.com
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without
* restriction, including without limitation the rights to use,
* copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the
* Software is furnished to do so, subject to the following
* conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY
 OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
* HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
* OTHER DEALINGS IN THE SOFTWARE.
*/
/*
* jsPlumb
*
* Title:jsPlumb 2.2.0
*
* Provides a way to visually connect elements on an HTML page, using SVG.
*
* This file contains utility functions that run in both browsers and headless.
*
* Copyright (c) 2010 - 2016 jsPlumb (hello@jsplumbtoolkit.com)
*
* http://jsplumbtoolkit.com
* http://github.com/sporritt/jsplumb
*
* Dual licensed under the MIT and GPL2 licenses.
*/
```

Found in path(s):

* /opt/cola/permits/136814187_1695303771.4737241/0/jsplumb-2-2-0-tar/jsPlumb-2.2.0.js

1.13 redux-thunk 2.1.0

1.13.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights

1.14 antlr 4.5.3

1.14.1 Available under license :

Bud1 r4.txt
antlr4.txtIlocblob < (

@
@ @ @

E

DSDB `
@ @
@

1.15 cookie 0.3.1

1.15.1 Available under license :

(The MIT License)

Copyright (c) 2012-2014 Roman Shtylman <shtylman@gmail.com>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR
 OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.16 xerces-j 2.9.1

1.16.1 Available under license :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<?xml version="1.0" encoding="iso-8859-1"?><HTML
xmlns="http://www.w3.org/1999/xhtml"><HEAD><TITLE>W3C Software License</TITLE>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">

<META content="MSHTML 6.00.2800.1400" name=GENERATOR></HEAD>
<BODY text=#000000 bgColor=#ffffff>
<H1>W3C<SUP>®</SUP> SOFTWARE NOTICE AND LICENSE</H1>
<H3 id=version><A
href="http://www.w3.org/Consortium/Legal/2002/copyright-software-
20021231">http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231</A></H3>
<P>This work (and included software, documentation such as READMEs, or other
related items) is being provided by the copyright holders under the following
license. By obtaining, using and/or copying this work, you (the licensee) agree
that you have read, understood, and will comply with the following terms and
conditions.</P>
<P>Permission to copy, modify, and distribute this software and its
documentation,
 with or without modification,&nbsp;for any purpose and without
fee or royalty is hereby granted, provided that you include the following on ALL
copies of the software and documentation or portions thereof, including
modifications:</P>
<OL>
 <LI>The full text of this NOTICE in a location viewable to users of the
 redistributed or derivative work.
 <LI>Any pre-existing intellectual property disclaimers, notices, or terms and
 conditions. If none exist, the <A
 href="http://www.w3.org/Consortium/Legal/2002/copyright-software-short-notice-20021231.html">W3C
 Software Short Notice</A> should be included (hypertext is preferred, text is
 permitted) within the body of any redistributed or derivative code.
 <LI>Notice of any changes or modifications to the files, including the date
 changes were made. (We recommend you provide URIs to the location from which
 the code is derived.) </LI></OL>
<P>THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
MAKE
 NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT
LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE
OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD
```

PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.</P>
<P>COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
DOCUMENTATION.</P>
<P>The name and trademarks of copyright holders may NOT be used in advertising
or publicity pertaining to the software without specific, written prior
permission. Title to copyright in this software and any associated documentation
will at all times remain with copyright holders.</P>
<P></P>
<P>__________________________________</P>
<P>This formulation of W3C's notice and license became active on December 31
2002. This version removes the copyright ownership notice such that this license
can be used with materials other than those
owned by the W3C, reflects that
ERCIM is now a host of the W3C, includes references to this specific dated
version of the license, and removes the ambiguous grant of "use". Otherwise,
this version is the same as the <A
href="http://www.w3.org/Consortium/Legal/copyright-software-19980720">previous
version</A> and is written so as to preserve the <A
href="http://www.gnu.org/philosophy/license-list.html#GPLCompatibleLicenses">Free
Software Foundation's assessment of GPL compatibility</A> and <A
href="http://www.opensource.org/licenses/W3C.php">OSI's certification</A> under
the <A href="http://www.opensource.org/docs/definition.php">Open Source
Definition</A>. Please see our <A
href="http://www.w3.org/Consortium/Legal/IPR-FAQ">Copyright FAQ</A> for common
questions about using materials from our site, including specific terms and
conditions for packages like libwww, Amaya, and Jigsaw. Other questions about
this notice can be directed to <A
href="mailto:site-policy@w3.org">site-policy@w3.org</A>.<BR>&nbsp;</P>
<ADDRESS>Joseph
Reagle &lt;<A
href="mailto:site-policy@w3.org">mailto:site-policy@w3.org</A> </ADDRESS>
<P>Last revised by Reagle $Date: 2004-04-06 13:20:26 -0400 (Tue, 06 Apr 2004) $</P></BODY></HTML>
Apache XML Commons Resolver
Copyright 2006 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation http://www.apache.org/

Portions of this code are derived from classes placed in the
public domain by Arbortext on 10 Apr 2000. See:
http://www.arbortext.com/customer_support/updates_and_technical_notes/catalogs/docs/README.htm
NOT PROVIDED
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<?xml version="1.0" encoding="iso-8859-1"?><HTML
xmlns="http://www.w3.org/1999/xhtml"><HEAD><TITLE>W3C Document License</TITLE>
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">

```
<META content="MSHTML 6.00.2800.1400" name=GENERATOR></HEAD>
<BODY text=#000000 bgColor=#ffffff>
<H1>W3C<SUP>®</SUP> DOCUMENT LICENSE</H1>
<H3 id=version><A
href="http://www.w3.org/Consortium/Legal/2002/copyright-documents-
20021231">http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231</A></H3>
<P>Public documents on the W3C site are provided by the copyright holders under
the following license. By using and/or copying this document, or the W3C
document from which this statement is linked, you (the licensee) agree that you
have read, understood, and will comply with the following terms and
conditions:</P>
<P>Permission to copy, and distribute the contents of this document, or the W3C
document from which this
 statement is linked, in any medium for any purpose and
without fee or royalty is hereby granted, provided that you include the
following on <I>ALL</I> copies of the document, or portions thereof, that you
use:</P>
<OL>
 <LI>A link or URL to the original W3C document.
 <LI>The pre-existing copyright notice of the original author, or if it doesn't
 exist, a notice (hypertext is preferred, but a textual representation is
 permitted) of the form: "Copyright © [$date-of-document] <A
 href="http://www.w3.org/">World Wide Web Consortium</A>, (<A
 href="http://www.lcs.mit.edu/">Massachusetts Institute of Technology</A>, <A
 href="http://www.ercim.org/">European Research Consortium for Informatics and
 Mathematics</A>, <A href="http://www.keio.ac.jp/">Keio University</A>). All
 Rights Reserved. <A
 href="http://www.w3.org/Consortium/Legal/2002/copyright-documents-
20021231">http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231</A>"

 <LI><EM>If it exists</EM>,
 the STATUS of the W3C document. </LI></OL>
<P>When space permits, inclusion of the full text of this <B>NOTICE</B> should
be provided. We request that authorship attribution be provided in any software,
documents, or other items or products that you create pursuant to the
implementation of the contents of this document, or any portion thereof.</P>
<P>No right to create modifications or derivatives of W3C documents is granted
pursuant to this license. However, if additional requirements (documented in the
<A href="http://www.w3.org/Consortium/Legal/IPR-FAQ">Copyright FAQ</A>) are
satisfied, the right to create modifications or derivatives is&nbsp;sometimes
granted by the W3C to individuals complying with those requirements.</P>
<P>THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO
REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED
TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,
NON-INFRINGEMENT, OR TITLE; THAT
 THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR
ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY
THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.</P>
```

```
<P>COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE
PERFORMANCE
OR IMPLEMENTATION OF THE CONTENTS THEREOF.</P>
<P>The name and trademarks of copyright holders may NOT be used in advertising
or publicity pertaining to this document or its contents without specific,
written prior permission. Title to copyright in this document will at all times
remain with copyright holders.</P>
<P></P>
<P>-------------------------------------------------------------------------</P>
<P>This formulation of W3C's notice and license became active on December 31
2002. This version removes the copyright ownership notice such that this license
can be used with materials other than those owned by the W3C, moves information

on style sheets, DTDs, and schemas to the <A
href="http://www.w3.org/Consortium/Legal/IPR-FAQ">Copyright FAQ</A>, reflects
that ERCIM is now a host of the W3C, includes references to this specific dated
version of the license, and removes the ambiguous grant of "use". See the <A
href="http://www.w3.org/Consortium/Legal/copyright-documents-19990405">older
formulation</A> for the policy prior to this date. Please see our <A
href="http://www.w3.org/Consortium/Legal/IPR-FAQ">Copyright FAQ</A> for common
questions about using materials from our site, such as the translating or
annotating specifications. Other questions about this notice can be directed to
<A href="mailto:site-policy@w3.org">site-policy@w3.org</A>.</P>
<P></P>
<ADDRESS>Joseph Reagle &lt;<A
href="mailto:site-policy@w3.org">mailto:site-policy@w3.org</A> </ADDRESS>
<P>Last revised by Reagle $Date: 2004-04-06 13:20:26 -0400 (Tue, 06 Apr 2004) $</P></BODY></HTML>
 <html>
<head>SAX LICENSE</head>
<body>
<p>This license applies to all interfaces and classes in the
org/xml/sax hierarchy.
</p>
<p>
<blockquote>
<em>This module, both source code and documentation, is in the
Public Domain, and comes with <strong>NO WARRANTY</strong>.</em>
See <a href='http://www.saxproject.org'>http://www.saxproject.org</a>
for further information.
</blockquote>
</p>
</body>
</html>
```

```
   =========================================================================
  ==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
  ==  Version 2.0, in this case for the Apache Xerces Java distribution. ==
  =========================================================================

  Apache Xerces Java
  Copyright 1999-2007 The Apache Software Foundation

  This product includes software developed at
  The Apache Software Foundation (http://www.apache.org/).

  Portions of this software were originally based on the following:
    - software copyright (c) 1999, IBM Corporation., http://www.ibm.com.
    - software copyright (c) 1999, Sun Microsystems., http://www.sun.com.
    - voluntary contributions made by Paul Eng on behalf of the
      Apache Software Foundation that were originally developed at iClick, Inc.,
      software copyright (c) 1999.

            Xerces Java Build Instructions
            ------------------------------

==========================================================================
```

==

Before
 building Xerces, you need the source package and tools
package available from the Apache XML Project's distribution
web page:

   http://xml.apache.org/dist/xerces-j/

Download both the Xerces-J-src.X.Y.Z.zip and Xerces-J-tools.X.Y.Z.zip
files for the appropriate Xerces release (where "X.Y.Z" is the version
number) and extract them in the same directory. If you are using Unix,
download the equivalent .tar.gz files instead of the .zip files.

You also need to have a Java Development Kit (JDK) version 1.2 (or higher) installed on your system. Before initiating any part of the build, set the JAVA_HOME environment variable to the installation directory of your JDK.

The Ant program is used to build everything in Xerces, including the documentation. This tool, and the others needed (besides the pre-requisite JDK) are contained within the tools package. To make building the packages easier, a Windows batch file and a Unix shell script
are included.

If you only want to compile the source code and make the JAR files, run the following command on Windows:

```
  build.bat jars
```

or from Unix (make sure that build.sh is executable):

```
  build.sh jars
```

This will compile all of the source code and generate the JAR files that are available as part of the binary package. After building, these files will be located in the build/ directory.

If you want to build everything, including the documentation, run the build batch file (or shell script) specifying the "all" target instead of "jars".

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the

origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===
== NOTICE file corresponding to section 4(d) of the Apache License, ==
== Version 2.0, in this case for the Apache Xalan Java distribution. ==
===

Apache Xalan (Xalan serializer)
Copyright 1999-2006 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software was originally based on the following:
- software copyright (c) 1999-2002, Lotus Development Corporation.,
  http://www.lotus.com.
- software copyright (c) 2001-2002, Sun Microsystems.,
  http://www.sun.com.
- software copyright (c) 2003, IBM Corporation.,
  http://www.ibm.com.

1.17 request 2.81.0

1.17.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean
 the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name)
 to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual
 or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to

You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by
combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and

assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall
any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims
asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.18 moment-duration-format 1.3.0

1.18.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 John Madhavan-Reese

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.19 asm 2.2.0

1.19.1 Available under license :

No license file was found, but licenses were detected in source scan.

ASM XML Adapter
Copyright (c) 2004, Eugene Kuleshov
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/asm-xml.dtd
No license file was found, but licenses were detected in source scan.

<!--
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2005 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/attrs/package.html
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/package.html
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/package.html
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/package.html
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/package.html
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/signature/package.html
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/package.html
No license file was found, but licenses were detected in source scan.

* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/InstructionAdapter.java
No license file was found, but licenses were detected in source scan.

/***
* ASM XML Adapter
* Copyright (c) 2004, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES,

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/SAXCodeAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/ASMContentHandler.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/Processor.java
*
/opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/SAXFieldAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/SAXClassAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/SAXAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/xml/package.html
No license file was found, but licenses were detected in source scan.

```
   only the GPL Version 2, indicate your decision by adding "[Contributor]
   elects to include this software in this distribution under the [CDDL or GPL
   Version 2] license."  If you don't indicate a single choice of license, a
   recipient has the option to distribute your version of this file under
   either the CDDL, the GPL Version 2 or to extend the choice of license to
   its licensees as provided above.  However, if you add GPL Version 2 code
   and therefore, elected the GPL Version 2 license, then the option applies
   only if
 the new code is made subject to such option by the copyright
   holder.

-->
```

Found in path(s):

* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/pom.xml

No license file was found, but licenses were detected in source scan.

```
/**
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2007 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/ASMifiable.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/Traceable.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/signature/SignatureVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/MethodNode.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/MultiANewArrayInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/CheckClassAdapter.java

* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/AbstractVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Attribute.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/LookupSwitchInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/RemappingSignatureAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/AbstractInsnNode.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/ASMifierAnnotationVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/ClassVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/CodeSizeEvaluator.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/Value.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/LocalVariableNode.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/TraceAbstractVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/Subroutine.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/SerialVersionUIDAdder.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/CheckMethodAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/ASMifierFieldVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/RemappingAnnotationAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Type.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/AnalyzerAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/RemappingClassAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-

jar/org/objectweb/asm/tree/InnerClassNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/TableSwitchGenerator.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/BasicValue.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/SmallSet.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/MemberNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/CheckAnnotationAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/AnnotationVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/Frame.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/ASMifierClassVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/RemappingFieldAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/AnalyzerException.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Item.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/ASMifierAbstractVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/TypeInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/FieldNode.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/StaticInitMerger.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/IincInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/BasicVerifier.java

* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/LocalVariablesSorter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/AnnotationNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/GeneratorAdapter.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/Analyzer.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/JumpInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/SourceValue.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/EmptyVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/MethodAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/LineNumberNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/ByteVector.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/Interpreter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/Remapper.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Frame.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/VarInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/TableSwitchInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/InsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/SimpleVerifier.java
*
 /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/ASMifierMethodVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/FrameNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/CheckFieldAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-

jar/org/objectweb/asm/tree/InsnList.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/TraceSignatureVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/TraceClassVisitor.java
*
/opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/ClassNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/TraceMethodVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/AnnotationWriter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/Method.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/IntInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/SimpleRemapper.java
*
/opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/util/TraceFieldVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/LabelNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/MethodInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/RemappingMethodAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/FieldInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/analysis/SourceInterpreter.java
*
/opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/TryCatchBlockNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/ClassAdapter.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/tree/LdcInsnNode.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/ClassWriter.java

* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/Opcodes.java
* /opt/cola/permits/160087105_1695311834.3746471/0/asm-all-repackaged-2-2-0-b21-sources-jar/org/objectweb/asm/commons/AdviceAdapter.java

1.20 xmlschema 2.0.3

1.20.1 Available under license :

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.21 bouncy-castle 1.55

1.21.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**

* The Bouncy Castle License
*
* Copyright (c) 2000-2015 The Legion Of The Bouncy Castle Inc. (http://www.bouncycastle.org)
* <p>
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software
* and associated documentation files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
* and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
* <p>
* The above copyright notice and this permission notice shall be included in all copies or substantial
* portions of the Software.
* <p>
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
* INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
* PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
* OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
* DEALINGS IN THE SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/160982470_1695312053.0469542/0/bcprov-jdk15on-1-55-sources-jar/org/bouncycastle/LICENSE.java
No license file was found, but licenses were detected in source scan.

public DomainParameters(BigInteger p, BigInteger g, BigInteger q, BigInteger j,

Found in path(s):
* /opt/cola/permits/160982470_1695312053.0469542/0/bcprov-jdk15on-1-55-sources-jar/org/bouncycastle/asn1/x9/DomainParameters.java
No license file was found, but licenses were detected in source scan.

/**
* Poly1305 message authentication code, designed by D. J. Bernstein.
* <p>
* Poly1305 computes a 128-bit (16 bytes) authenticator, using a 128 bit nonce and a 256 bit key
* consisting of a 128 bit key applied to an underlying cipher, and a 128 bit key (with 106
* effective key bits) used in the authenticator.
* <p>
* The polynomial calculation in this implementation is adapted from the public domain <a
* href="https://github.com/floodyberry/poly1305-donna">poly1305-donna-unrolled</a> C implementation
* by Andrew M (@floodyberry).

* @see Poly1305KeyGenerator
*/

Found in path(s):
* /opt/cola/permits/160982470_1695312053.0469542/0/bcprov-jdk15on-1-55-sources-jar/org/bouncycastle/crypto/macs/Poly1305.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/160982470_1695312053.0469542/0/bcprov-jdk15on-1-55-sources-jar/org/bouncycastle/pqc/math/ntru/euclid/test/IntEuclideanTest.java
No license file was found, but licenses were detected in source scan.

// now keep space is free to use

Found in path(s):
* /opt/cola/permits/160982470_1695312053.0469542/0/bcprov-jdk15on-1-55-sources-jar/org/bouncycastle/pqc/crypto/gmss/GMSSPrivateKeyParameters.java

1.22 commons-pool 2.4.2

1.22.1 Available under license :

Apache Commons Pool
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

APACHE COMMONS POOL DERIVATIVE WORKS:

The LinkedBlockingDeque implementation is based on an implementation written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at http://creativecommons.org/licenses/publicdomain

1.23 commons-collections 3.2.2

1.23.1 Available under license :

Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

1.24 commons-logging 1.2

1.24.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

Apache Commons Logging
Copyright 2003-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.25 google-protobuf 3.2.0

1.25.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/GeneratedMessageLite.java
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcUtil.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapFieldLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ByteBufferWriter.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/wrappers.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractMessage.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapEntry.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnknownFieldSetLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/empty.proto
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Service.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/GeneratedMessage.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcController.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyStringArrayList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnsafeUtil.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Utf8.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/FloatArrayList.java

* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/DynamicMessage.java
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Descriptors.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyFieldLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExperimentalApi.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageOrBuilder.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnknownFieldSet.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractMessageLite.java
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormat.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractParser.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/DoubleArrayList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LongArrayList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RopeByteString.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ProtocolMessageEnum.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/NioByteString.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UninitializedMessageException.java
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageReflection.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ByteOutput.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/CodedInputStream.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/any.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageLiteOrBuilder.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-

jar/com/google/protobuf/CodedOutputStream.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/GeneratedMessageV3.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyStringList.java
*
/opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Extension.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/source_context.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ProtobufArrayList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnsafeByteOperations.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapEntryLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionRegistry.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/InvalidProtocolBufferException.java
*
/opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/SingleFieldBuilderV3.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RepeatedFieldBuilder.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionRegistryFactory.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionRegistryLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapField.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormatParseInfoTree.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ByteString.java
*
/opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ProtocolStringList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MutabilityOracle.java

* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/BlockingService.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/IntArrayList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Message.java
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/compiler/plugin.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/struct.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/FieldSet.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/api.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/type.proto
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractProtobufList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/duration.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyField.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionLite.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/WireFormat.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Internal.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcChannel.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Parser.java
*
 /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/BooleanArrayList.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormatEscaper.java
* /opt/cola/permits/163181375_1695313992.1002758/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ServiceException.java

1.26 mime-pull 1.9.3

1.26.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 1997-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/Header.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/MIMEMessage.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/MIMEConfig.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/PropUtil.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/BASE64DecoderStream.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/ChunkInputStream.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/FactoryFinder.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/MIMEParsingException.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/UUDecoderStream.java
*
 /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/CleanUpExecutorFactory.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/DecodingException.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/LineInputStream.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/FinalArrayList.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/Chunk.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/WeakDataFile.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/DataFile.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/InternetHeaders.java
*
 /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/QPDecoderStream.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/ASCIIUtility.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/Data.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/MIMEEvent.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/FileData.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/MimeUtility.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-

jar/org/jvnet/mimepull/MIMEParser.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/MIMEPart.java
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception as provided
 by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to
 such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/TempFiles.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-

jar/org/jvnet/mimepull/MemoryData.java
* /opt/cola/permits/163338568_1695314318.1083882/0/mimepull-1-9-3-sources-jar/org/jvnet/mimepull/DataHead.java

1.27 resourcelocator 1.0.1

1.27.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found

in path(s):

* /opt/cola/permits/163339037_1695314327.1315129/0/osgi-resource-locator-1-0-1-sources-zip/org/glassfish/hk2/osgiresourcelocator/ResourceFinderImpl.java
* /opt/cola/permits/163339037_1695314327.1315129/0/osgi-resource-locator-1-0-1-sources-zip/org/glassfish/hk2/osgiresourcelocator/ServiceLoaderImpl.java
* /opt/cola/permits/163339037_1695314327.1315129/0/osgi-resource-locator-1-0-1-sources-zip/org/glassfish/hk2/osgiresourcelocator/ServiceLoader.java
* /opt/cola/permits/163339037_1695314327.1315129/0/osgi-resource-locator-1-0-1-sources-zip/org/glassfish/hk2/osgiresourcelocator/ResourceFinder.java
* /opt/cola/permits/163339037_1695314327.1315129/0/osgi-resource-locator-1-0-1-sources-zip/org/glassfish/hk2/osgiresourcelocator/Activator.java

1.28 spring-tx 3.0.5

1.28.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2007 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/NoRollbackRuleAttribute.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2010 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
```

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/work/glassfish/GlassFishWorkManagerTaskExecutor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/SimpleTransactionFactory.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/AnnotationTransactionAttributeSource.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/UserTransactionAdapter.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/AbstractPlatformTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAspectUtils.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/NameMatchTransactionAttributeSource.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAspectSupport.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/WebLogicJtaTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/WebSphereUowTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/MatchAlwaysTransactionAttributeSource.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/Transactional.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAttributeSource.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/AbstractFallbackTransactionAttributeSource.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/RuleBasedTransactionAttribute.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/JtaTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/ManagedTransactionAdapter.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/MethodMapTransactionAttributeSource.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-

jar/org/springframework/transaction/support/TransactionSynchronizationManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/work/jboss/JBossWorkManagerUtils.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/endpoint/AbstractMessageEndpointFactory.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/DefaultTransactionAttribute.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/OC4JJtaTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/ConnectionSpecConnectionFactoryAdapter.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/work/jboss/JBossWorkManagerTaskExecutor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionInterceptor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/CompositeTransactionAttributeSource.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/annotation/PersistenceExceptionTranslationPostProcessor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/support/PersistenceExceptionTranslationInterceptor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionDefinition.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/IllegalTransactionStateException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionException.java

*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/UnexpectedRollbackException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/DataRetrievalFailureException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/support/DaoSupport.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/OptimisticLockingFailureException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/CleanupFailureDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionSuspensionNotSupportedException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/IncorrectUpdateSemanticsDataAccessException.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/InvalidTimeoutException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/DataAccessResourceFailureException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/IncorrectResultSizeDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/PermissionDeniedDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/UncategorizedDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/CannotCreateTransactionException.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/InvalidIsolationLevelException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/CannotSerializeTransactionException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/ConcurrencyFailureException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionUsageException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/support/PersistenceExceptionTranslator.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/PessimisticLockingFailureException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/NestedTransactionNotSupportedException.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/TypeMismatchDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/InvalidDataAccessResourceUsageException.java

* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/HeuristicCompletionException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/DeadlockLoserDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/EmptyResultDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/DataAccessException.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/DataIntegrityViolationException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAttributeEditor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/CannotAcquireLockException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/NoTransactionException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionTimedOutException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/InvalidDataAccessApiUsageException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/InvalidResultSetAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/object/MappingCommAreaOperation.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/support/CommAreaRecord.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-

jar/org/springframework/jca/cci/connection/ConnectionHolder.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/CannotGetCciConnectionException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/RecordCreator.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/object/MappingRecordOperation.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/DuplicateKeyException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionSynchronizationAdapter.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/AbstractTransactionStatus.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/ResourceHolderSynchronization.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/config/TxNamespaceHandler.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/support/LocalConnectionFactoryBean.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/DelegatingTransactionAttribute.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/SpringTransactionAnnotationParser.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/SmartTransactionObject.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/config/AnnotationDrivenBeanDefinitionParser.java

* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionCallback.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionProxyFactoryBean.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/DefaultTransactionStatus.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/ConnectionFactoryUtils.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionOperations.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/support/ResourceAdapterFactoryBean.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/JtaAfterCompletionSynchronization.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/config/TxAdviceBeanDefinitionParser.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/SimpleTransactionStatus.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/JtaTransactionObject.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/Isolation.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/CallbackPreferringPlatformTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionSynchronization.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAttribute.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/context/ResourceAdapterApplicationContext.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/DelegatingTransactionDefinition.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/work/WorkManagerTaskExecutor.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionCallbackWithoutResult.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/Propagation.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionSynchronizationUtils.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/TransactionFactory.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/support/DataAccessUtils.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-

jar/org/springframework/jca/work/SimpleTaskWorkManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/endpoint/GenericMessageEndpointManager.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionStatus.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/config/JtaTransactionManagerBeanDefinitionParser.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/Ejb3TransactionAnnotationParser.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/TransactionTemplate.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/RollbackRuleAttribute.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/support/SimpleBootstrapContext.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/ResourceTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/DelegatingConnectionFactory.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/jta/SpringJtaSynchronizationAdapter.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/TransactionSystemException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/annotation/TransactionAnnotationParser.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/context/BootstrapContextAwareProcessor.java

* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/context/BootstrapContextAware.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/work/DelegatingWork.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/endpoint/GenericMessageEndpointFactory.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/annotation/PersistenceExceptionTranslationAdvisor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/CciLocalTransactionManager.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/context/SpringContextResourceAdapter.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAttributeSourcePointcut.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/TransientDataAccessResourceException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAttributeSourceAdvisor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/DefaultTransactionDefinition.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/support/ChainedPersistenceExceptionTranslator.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/CciOperations.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/InteractionCallback.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-

jar/org/springframework/dao/RecoverableDataAccessException.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/RecordExtractor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/support/CciDaoSupport.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/TransactionAttributeSourceEditor.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/ResourceHolderSupport.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/TransientDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/NonTransientDataAccessResourceException.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/PlatformTransactionManager.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/dao/NonTransientDataAccessException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/ConnectionCallback.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/SingleConnectionFactory.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/TransactionAwareConnectionFactoryProxy.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/support/ResourceHolder.java
*
/opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/core/CciTemplate.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/interceptor/BeanFactoryTransactionAttributeSourceAdvisor.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/CciOperationNotSupportedException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/transaction/SavepointManager.java
*
 /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/connection/NotSupportedRecordFactory.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/CannotCreateRecordException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/object/SimpleRecordOperation.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/RecordTypeNotSupportedException.java
* /opt/cola/permits/163497771_1695314598.5855906/0/spring-tx-3-0-5-release-sources-11-jar/org/springframework/jca/cci/object/EisOperation.java

1.29 expression 3.0.5

1.29.1 Available under license :

No license file was found, but licenses were detected in source scan.

() [] {}

Found in path(s):
* // brackets must be in pairs
No license file was found, but licenses were detected in source scan.

" +

Found in path(s):
* "' for expression starting at character " + prefixIndex + "
No license file was found, but licenses were detected in source scan.

"foo${expr0}bar${expr1}". The static

Found in path(s):
* * number of expressions all contained in "${...}" markers. For instance
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2010 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/AstUtils.java
No license file was found, but licenses were detected in source scan.

case ']':case ')':

Found in path(s):
* case '}'
No license file was found, but licenses were detected in source scan.

Found in path(s):
* // Chew on the expression text - relying on the rules
* case '"'
* case '\''
No license file was found, but licenses were detected in source scan.

case '[': case '(':

Found in path(s):
* case '{'
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2004-2009 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

Found in path(s):
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ParserContext.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelParseException.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelEvaluationException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpOr.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpAnd.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/TypeCode.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/MethodReference.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/EvaluationException.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorInstanceof.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/SpelExpression.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpModulus.java

*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpMultiply.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/OperatorOverloader.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/StringLiteral.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpLT.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ExpressionState.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Operator.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/Tokenizer.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ExpressionParser.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpPlus.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorNot.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/SpelNodeImpl.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/LongLiteral.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/MethodExecutor.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/MethodFilter.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardOperatorOverloader.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/CompoundExpression.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpGT.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/Token.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/PropertyAccessor.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypeComparator.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/VariableReference.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/InternalSpelExpressionParser.java

* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Assign.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorPower.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectionHelper.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/TokenKind.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ParseException.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ExpressionException.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/NullLiteral.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpEQ.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Ternary.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpMinus.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/Operation.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/LiteralExpression.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardTypeLocator.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/AccessException.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ConstructorExecutor.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/CompositeStringExpression.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelMessage.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/BooleanLiteral.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorBetween.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Elvis.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypeLocator.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpDivide.java

* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Identifier.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpGE.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelParserConfiguration.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpLE.java
*
/opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/QualifiedIdentifier.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/SpelExpressionParser.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/BooleanTypedValue.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardTypeComparator.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelNode.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Literal.java
*
/opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/InternalParseException.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpNE.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/RealLiteral.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardEvaluationContext.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/TemplateAwareExpressionParser.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorMatches.java
*
/opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/TypeReference.java
No license file was found, but licenses were detected in source scan.

square brackets [] round brackets () and curly brackets {}

Found in path(s):
* * Bracket is used to describe
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2010 the original author or authors.
*

Found in path(s):
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypeConverter.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveMethodResolver.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Selection.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/FormatHelper.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ConstructorResolver.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/ConstructorReference.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/Expression.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Indexer.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveConstructorResolver.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/InlineList.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypedValue.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/BeanReference.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/BeanResolver.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ExpressionInvocationTargetException.java
*
 /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardTypeConverter.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-

jar/org/springframework/expression/common/TemplateParserContext.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/PropertyOrFieldReference.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveMethodExecutor.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/EvaluationContext.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/MethodResolver.java
*
/opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/ExpressionUtils.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/IntLiteral.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectivePropertyAccessor.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/FunctionReference.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Projection.java
* /opt/cola/permits/163553345_1695314620.0910468/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveConstructorExecutor.java
No license file was found, but licenses were detected in source scan.

//www.apache.org/licenses/LICENSE-2.0

Found in path(s):
* * http

1.30 spring-asm 3.0.5

1.31 spring-expression 3.0.5

1.31.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/TokenKind.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/LongLiteral.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/EvaluationException.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpNE.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/RealLiteral.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Literal.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/MethodFilter.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpGE.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypeComparator.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/QualifiedIdentifier.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpPlus.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorNot.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/BooleanLiteral.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/SpelExpressionParser.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardTypeComparator.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/SpelExpression.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/LiteralExpression.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ExpressionException.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-

jar/org/springframework/expression/spel/ast/Operator.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/InternalSpelExpressionParser.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectionHelper.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/CompoundExpression.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardTypeLocator.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/BooleanTypedValue.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/AccessException.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/TypeReference.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/Tokenizer.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpLT.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpMultiply.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ConstructorExecutor.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpOr.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/VariableReference.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Elvis.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypeLocator.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpAnd.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/OperatorOverloader.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Assign.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorInstanceof.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/CompositeStringExpression.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorPower.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-

jar/org/springframework/expression/spel/support/StandardOperatorOverloader.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Identifier.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/NullLiteral.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/StringLiteral.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/Operation.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/InternalParseException.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ExpressionParser.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/MethodReference.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Ternary.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/TemplateAwareExpressionParser.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelMessage.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpDivide.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpMinus.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorMatches.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/standard/Token.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ExpressionState.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OperatorBetween.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/PropertyAccessor.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpLE.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardEvaluationContext.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/SpelNodeImpl.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ParseException.java
*

/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelNode.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/MethodExecutor.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpGT.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpEQ.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/OpModulus.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/TypeCode.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelParserConfiguration.java
No license file was found, but licenses were detected in source scan.

() [] {}

Found in path(s):
* // brackets must be in pairs
No license file was found, but licenses were detected in source scan.

" +

Found in path(s):
* "' for expression starting at character " + prefixIndex + "
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelEvaluationException.java

* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/SpelParseException.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ParserContext.java
No license file was found, but licenses were detected in source scan.

"foo${expr0}bar${expr1}". The static

Found in path(s):
* * number of expressions all contained in "${...}" markers. For instance
No license file was found, but licenses were detected in source scan.

case ']':case ')':

Found in path(s):
* case '}'
No license file was found, but licenses were detected in source scan.

case '[': case '(':

Found in path(s):
* case '{'
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/EvaluationContext.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectivePropertyAccessor.java
*

/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/ExpressionInvocationTargetException.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/ConstructorReference.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/BeanReference.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/Expression.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/StandardTypeConverter.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/ExpressionUtils.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/FormatHelper.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/IntLiteral.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/MethodResolver.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Selection.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypeConverter.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/FunctionReference.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Projection.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/BeanResolver.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/PropertyOrFieldReference.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveMethodExecutor.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveConstructorExecutor.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/InlineList.java
*
/opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveConstructorResolver.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/common/TemplateParserContext.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/support/ReflectiveMethodResolver.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/TypedValue.java
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-

jar/org/springframework/expression/ConstructorResolver.java
*
 /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/Indexer.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* // Chew on the expression text - relying on the rules
* case '\'
* case ''
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163591223_1695314643.1209786/0/org-springframework-expression-sources-3-0-5-release-2-jar/org/springframework/expression/spel/ast/AstUtils.java
No license file was found, but licenses were detected in source scan.

square brackets [] round brackets () and curly brackets {}

Found in path(s):
* * Bracket is used to describe
No license file was found, but licenses were detected in source scan.

//www.apache.org/licenses/LICENSE-2.0

Found in path(s):
* *   http

1.32 spring-framework 3.0.5

1.32.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2010 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToArrayConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/GenericConversionService.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/LinkedMultiValueMap.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/AbstractStaxXMLReader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/support/SerializingConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/AbstractResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/ResourcePatternUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/TypeUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/StringUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/ConverterNotFoundException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/Resource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-

jar/org/springframework/util/xml/AbstractStaxContentHandler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/GenericTypeResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/SimpleNamespaceContext.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/Deserializer.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/VfsResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/WeakReferenceMonitor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxEventXMLReader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/XMLEventStreamReader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/SystemPropertyUtils.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/ConversionFailedException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ObjectToCollectionConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ExceptionDepthComparator.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/DefaultDeserializer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/TypeDescriptor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/VfsPatternUtils.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/CachingMetadataReaderFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/CollectionFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/PathMatchingResourcePatternResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/AnnotationMetadataReadingVisitor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/AntPathMatcher.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/support/SerializationFailedException.java
*

/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxStreamContentHandler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxEventContentHandler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/CollectionToStringConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/LocalVariableTableParameterNameDiscoverer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/GenericCollectionTypeResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/StopWatch.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/ResourceUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/SimpleAliasRegistry.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToCollectionConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/ResourceArrayPropertyEditor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/CollectionUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/ReflectionUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ObjectToObjectConverter.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/PropertyPlaceholderHelper.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/PathMatcher.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/XMLEventStreamWriter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/Serializer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/SerializationUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxStreamXMLReader.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/ResourceEditor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/MapToMapConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/CollectionToCollectionConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-

jar/org/springframework/core/convert/support/ConversionServiceFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/ConversionException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ConversionUtils.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/AbstractFileResolvingResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/DefaultSerializer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/serializer/support/DeserializingConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/annotation/AnnotationUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/ClassPathResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/VfsUtils.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/ClassUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/MethodParameter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/FileSystemUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/PropertyTypeDescriptor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/comparator/InvertibleComparator.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/comparator/NullSafeComparator.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ControlFlow.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/PropertiesPersister.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ErrorCoded.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-

jar/org/springframework/core/type/filter/AssignableTypeFilter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/InputStreamSource.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/CompositeIterator.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/InputStreamResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/AttributeAccessor.java
*

/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/SpringVersion.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/SimpleSaxErrorHandler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/PropertiesLoaderUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ConstantException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/annotation/AnnotationAwareOrderComparator.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ParameterNameDiscoverer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/LocalizedResourceHelper.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/style/ValueStyler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/MetadataReaderFactory.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/filter/AspectJTypeFilter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/filter/TypeFilter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/FileSystemResourceLoader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/ResourceLoader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/filter/RegexPatternTypeFilter.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/filter/AbstractTypeHierarchyTraversingFilter.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/ContextResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/TaskTimeoutException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/XmlValidationModeDetector.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/StringValueResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/ResourcePatternResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/ByteArrayResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/SmartClassLoader.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/EncodedResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/style/StylerUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/filter/AbstractClassTestingTypeFilter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/PatternMatchUtils.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2007 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/Assert.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/Log4jConfigurer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/DecoratingClassLoader.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/FileSystemResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/TaskExecutor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/FileCopyUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/comparator/BooleanComparator.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/NamedThreadLocal.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ConfigurableObjectInputStream.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/CommonsLogWriter.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/TaskRejectedException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/DomUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/DescriptiveResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/DefaultPropertiesPersister.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-

jar/org/springframework/util/xml/SimpleTransformErrorListener.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/comparator/ComparableComparator.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/ConcurrencyThrottleSupport.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/style/ToStringCreator.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/style/DefaultToStringStyler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/AutoPopulatingList.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/Constants.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/comparator/CompoundComparator.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/support/PropertiesLoaderSupport.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/NestedExceptionUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ControlFlowFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/style/ToStringStyler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/style/DefaultValueStyler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/NamedInheritableThreadLocal.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/InfrastructureProxy.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/AttributeAccessorSupport.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/PrioritizedParameterNameDiscoverer.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/AliasRegistry.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/TransformerUtils.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/UrlResource.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToNumberConverterFactory.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/MultiValueMap.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/NumberToNumberConverterFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/OverridingClassLoader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/ErrorHandler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/AnnotationMetadata.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/SimpleAsyncTaskExecutor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToLocaleConverter.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxResult.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/StaticLabeledEnumResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/MethodMetadata.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/SimpleMetadataReaderFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/NestedCheckedException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/converter/ConverterRegistry.java
*
 /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ArrayToArrayConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToBooleanConverter.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/CollectionToArrayConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/CharacterToNumberFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/DigestUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/AbstractGenericLabeledEnum.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ArrayToCollectionConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/AbstractXMLReader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/filter/AnnotationTypeFilter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/NumberToCharacterConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/IdToEntityConverter.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/ConversionService.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ConvertingPropertyEditorAdapter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/ObjectUtils.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToCharacterConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/StringCodedLabeledEnum.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/Conventions.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/ShortCodedLabeledEnum.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/CachingMapDecorator.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/DomContentHandler.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ArrayToStringConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/AbstractXMLStreamReader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/support/ExecutorServiceAdapter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/SyncTaskExecutor.java
*

/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/LinkedCaseInsensitiveMap.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/support/TaskExecutorAdapter.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/xml/StaxSource.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/StaticLabeledEnum.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/AntPathStringMatcher.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/SimpleMetadataReader.java

*

/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/ConcurrentMap.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/converter/Converter.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/DefaultResourceLoader.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/OrderComparator.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/JdkVersion.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/CustomizableThreadCreator.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/NumberUtils.java

*

/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/io/ClassRelativeResourceLoader.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ObjectToArrayConverter.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/LetterCodedLabeledEnum.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/util/MethodInvoker.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/ClassMetadata.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/converter/ConverterFactory.java

*

/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/AbstractLabeledEnum.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/support/ConcurrentExecutorAdapter.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/PriorityOrdered.java

* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-

jar/org/springframework/core/convert/support/StringToEnumConverterFactory.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/ClassMetadataReadingVisitor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/AbstractCachingLabeledEnumResolver.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/NestedIOException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/MetadataReader.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/AnnotationAttributesReadingVisitor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/converter/ConditionalGenericConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/StandardMethodMetadata.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/LabeledEnum.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ObjectToStringConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/task/AsyncTaskExecutor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/classreading/MethodMetadataReadingVisitor.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/BridgeMethodResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/enums/LabeledEnumResolver.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/converter/GenericConverter.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/StandardAnnotationMetadata.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/StringToPropertiesConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/PropertiesToStringConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/type/StandardClassMetadata.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/annotation/Order.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/ArrayToObjectConverter.java
*
/opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/convert/support/CollectionToObjectConverter.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-

jar/org/springframework/core/NestedRuntimeException.java
* /opt/cola/permits/163592078_1695314645.2949424/0/org-springframework-core-sources-3-0-5-release-2-jar/org/springframework/core/Ordered.java

1.33 jcip-annotation 1.0-1

1.33.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 Stephen Connolly.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/163766248_1695314659.254782/0/jcip-annotations-1-0-1-sources-zip/net/jcip/annotations/GuardedBy.java
* /opt/cola/permits/163766248_1695314659.254782/0/jcip-annotations-1-0-1-sources-zip/net/jcip/annotations/ThreadSafe.java
* /opt/cola/permits/163766248_1695314659.254782/0/jcip-annotations-1-0-1-sources-zip/net/jcip/annotations/NotThreadSafe.java
*
 /opt/cola/permits/163766248_1695314659.254782/0/jcip-annotations-1-0-1-sources-zip/net/jcip/annotations/Immutable.java

1.34 json-java 20140107

1.34.1 Available under license :

No license file was found, but licenses were detected in source scan.

<p class="legalCopy"><small>Copyright &#169; 2014. All Rights Reserved.</small></p>

Found in path(s):
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/XMLTokener.html

* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONML.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/BitReader.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/HTTPTokener.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/package-tree.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/HTTPTokener.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/PostMortem.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/HTTP.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/Kim.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/constant-values.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/Compressor.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/None.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/CookieList.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/XML.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/HTTP.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONObject.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONString.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/Decompressor.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/None.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/JSONzip.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/package-use.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONML.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/Property.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/BitWriter.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/Compressor.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/CookieList.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/XMLTokener.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/overview-tree.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/CDL.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONObject.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONStringer.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/package-tree.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/BitOutputStream.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-

use/JSONException.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONTokener.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/BitWriter.html
*
/opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/index-all.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONArray.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/JSONzip.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/Property.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/package-summary.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/BitInputStream.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONString.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONException.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/help-doc.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONWriter.html
*
/opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/BitInputStream.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/package-summary.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/XML.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/serialized-form.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/CDL.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONTokener.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/BitReader.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/package-use.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/BitOutputStream.html
*
/opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/Kim.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/Decompressor.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONWriter.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/Cookie.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/Huff.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/JSONStringer.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/JSONArray.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/class-use/Huff.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/class-use/Cookie.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/overview-summary.html
*
/opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/org/json/zip/PostMortem.html
* /opt/cola/permits/163783015_1695314665.9343166/0/json-20140107-javadoc-jar/deprecated-list.html

1.35 swagger 1.5.0

1.36 react-router-dom 4.1.1

1.36.1 Available under license :

1.37 redux 3.4.1

1.37.1 Available under license :

The laws of most jurisdictions throughout the world automatically confer exclusive Copyright and Related Rights (defined below) upon the creator and subsequent owner(s) (each and all, an "owner") of an original work of authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. These owners may contribute to the Commons to promote the ideal of a free culture and the further production of creative, cultural and scientific works, or to gain reputation or greater distribution for their Work in part through the use and efforts of others.

For these and/or other purposes and motivations, and without any expectation of additional consideration or compensation, the person associating CC0 with a Work (the "Affirmer"), to the extent that he or she is an owner of Copyright and Related Rights in the Work, voluntarily elects to apply CC0 to the Work and publicly distribute the Work under its terms, with knowledge of his or her Copyright and Related Rights in the Work and the meaning and intended legal effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be protected by copyright and related or neighboring rights ("Copyright and Related Rights"). Copyright and Related Rights include, but are not limited to, the following:

i. the right to reproduce, adapt, distribute, perform, display, communicate, and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout
the world
based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall
not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever,
including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned, surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or warranties
of any kind concerning the Work, express, implied,
statutory or otherwise,
including without limitation warranties of title, merchantability, fitness
for a particular purpose, non infringement, or the absence of latent or
other defects, accuracy, or the present or absence of errors, whether or not
discoverable, all to the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons
that may apply to the Work or any use thereof, including without limitation
any person's Copyright and Related Rights in the Work. Further, Affirmer
disclaims responsibility for obtaining any necessary consents, permissions
or other rights required for any use of the Work.

d. Affirmer understands and acknowledges that Creative Commons is not a
party to this document and has no duty or obligation with respect to this
CC0 or use of the Work.

For more information, please see
<http://creativecommons.org/publicdomain/zero/1.0/>
The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.38 redux-act 1.3.2

1.38.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.39 highcharts-no-data-to-display 0.1.2

1.40 rc-table 5.2.2

1.40.1 Available under license :

1.41 beam 0.6.0

1.41.1 Available under license :

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

1.42 fst 2.52

1.42.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the

preferred form for making modifications, including
but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name)
to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual
or Legal Entity on behalf
of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby
grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free,
irrevocable (except as stated in this section) patent license to make, have
made, use, offer to sell, sell, import, and otherwise transfer the Work, where
such license applies only to those patent claims licensable by such Contributor
that are necessarily infringed by their Contribution(s) alone or by
combination
of their Contribution(s) with the Work to which such Contribution(s) was
submitted. If You institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work or a
Contribution incorporated within the Work constitutes direct or contributory
patent infringement, then any patent licenses granted to You under this License
for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof
in any medium, with or without modifications, and in Source or Object form,
provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of
this License; and
You must cause any modified files to carry prominent notices stating that You
changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute,
all copyright, patent, trademark,
and attribution notices from the Source form
of the Work, excluding those notices that do not pertain to any part of the
Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any
Derivative Works that You distribute must include a readable copy of the
attribution notices contained within such NOTICE file, excluding those notices
that do not pertain to any part of the Derivative Works, in at least one of the
following places: within a NOTICE text file distributed as part of the
Derivative Works; within the Source form or documentation, if provided along
with the Derivative Works; or, within a display generated by the Derivative
Works, if and wherever such third-party notices normally appear. The contents of
the NOTICE file are for informational purposes only and do not modify the
License. You may add Your own attribution notices within Derivative Works that
You distribute, alongside or as an addendum to the NOTICE text from the Work,
provided
that such additional attribution notices cannot be construed as
modifying the License.
You may add Your own copyright statement to Your modifications and may provide
additional or different license terms and conditions for use, reproduction, or
distribution of Your modifications, or for any such Derivative Works as a whole,
provided Your use, reproduction, and distribution of the Work otherwise complies
with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your

sole responsibility, not on behalf of any other Contributor, and only if You
agree to indemnify, defend, and hold each Contributor harmless for any liability
incurred by, or
claims asserted against, such Contributor by reason of your
accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate
notice, with the fields enclosed by brackets "[]" replaced with your own
identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also
recommend that a file or class name and description of purpose be included on
the same "printed page" as the copyright notice for easier identification within
third-party archives.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

1.43 tinkerpop-gremlin 3.2

1.43.1 Available under license :

Apache TinkerPop (incubating)
Copyright 2015-2016 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache Commons Lang (AL ASF)

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

Apache Groovy (AL ASF)

This product includes/uses ANTLR (http://www.antlr2.org/)
developed by Terence Parr 1989-2006

This product bundles icons from the famfamfam.com silk icons set
http://www.famfamfam.com/lab/icons/silk/
Licensed under the Creative Commons Attribution Licence v2.5
http://creativecommons.org/licenses/by/2.5/

Apache
 Ivy (AL ASF)

Portions of Ivy were originally developed by
Jayasoft SARL (http://www.jayasoft.fr/)
and are licensed to the Apache Software Foundation under the
"Software Grant License Agreement"

SSH and SFTP support is provided by the JCraft JSch package,
which is open source software, available under
the terms of a BSD style license.
The original software and related information is available
at http://www.jcraft.com/jsch/.

JavaTuples

Copyright (c) 2010, The JAVATUPLES team (http://www.javatuples.org)

Metrics

Copyright 2010-2013 Coda Hale and Yammer, Inc., 2014-2015 Dropwizard Team

This
 product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

   Written by Doug Lea with assistance from members of JCP JSR-166
   Expert Group and released to the public domain, as explained at
   http://creativecommons.org/publicdomain/zero/1.0/

Netty
----

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Esoteric Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) Nicolas Gallagher and Jonathan Neal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

==
MIT Licenses
==

The Apache TinkerPop project bundles the following components under the MIT License:

normalize.css 2.1.2 (http://necolas.github.io/normalize.css/) - for details, see licenses/normalize

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
BSD-style Licenses
==

The Apache TinkerPop project bundles the following components under the BSD License:

jcabi-log (com.jcabi:jcabi-log:0.14 - http://www.jcabi.com/jcabi-log) - for details, see licenses/jcabi-log
jcabi-manifests 1.1 (com.jcabi:jcabi-manifests:1.1 - http://www.jcabi.com/jcabi-manifests) - for details, see licenses/jcabi-manifests
JLine (jline:jline:2.12 - https://github.com/jline/jline2) - for details, see licenses/jline2
Kryo (com.esotericsoftware:kryo-shaded:3.0.1
- https://github.com/EsotericSoftware/kryo)
- shaded in gremlin-shaded to org.apache.tinkerpop.shaded.kryo
- for details, see licenses/kryo
minlog (com.esotericsoftware:minlog:1.3.0 - https://github.com/EsotericSoftware/minlog)
- shaded in gremlin-shaded to org.apache.tinkerpop.shaded.minlog

- for details, see licenses/minlog

==
MIT Licenses
==

The Apache TinkerPop project bundles the following components under the MIT License:

JCL 1.1.1 implemented over SLF4J (org.slf4j:jcl-over-slf4j:1.7.12 - http://www.slf4j.org) - for details, see licenses/slf4j
SLF4J API Module (org.slf4j:slf4j-api:1.7.12 - http://www.slf4j.org) - for details, see licenses/slf4j
SLF4J LOG4J-12 Binding (org.slf4j:slf4j-log4j12:1.7.12 - http://www.slf4j.org) - for details, see licenses/slf4j
Foundation stylesheet
for CodeRay (http://foundation.zurb.com) - for details, see licenses/foundation
normalize.css 2.1.2 (http://necolas.github.io/normalize.css/) - for details, see licenses/normalize

==
Other Licenses
==

The Apache TinkerPop project bundles the following components under the ISC License:

jBCrypt (com.github.jeremyh:jBCrypt:0.4 - https://github.com/jeremyh/jBCrypt) - for details, see licenses/jbcrypt

Apache TinkerPop (incubating)

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Activiti

Activiti BPM Platform
Copyright 2010-2014 Alfresco Software, Ltd.

gremlin-core/src/main/java/org/apache/tinkerpop/gremlin/structure/io/graphml/GraphMLWriterHelper.java
contains DelegatingXMLStreamWriter.java and IndentingXMLStreamWriter.java from
modules/activiti-bpmn-converter/src/main/java/org/activiti/bpmn/converter/
Apache TinkerPop is an effort undergoing incubation at The Apache Software Foundation (ASF),
sponsored by the Apache Incubator. Incubation is required of all newly accepted projects until
a further review indicates that the infrastructure, communications, and decision making process
have stabilized in a manner consistent with other successful ASF projects. While incubation
status is not necessarily a reflection of the completeness or stability of the code, it does
indicate that the project has yet to be fully endorsed by the ASF. Apache TinkerPop is distributed
under the Apache License v2.0.
Title: gmetric4j

Copyright:

Copyright (C) 2012 Daniel Pocock <daniel@pocock.com.au>
Copyright (c) 2008-2011 Jasper Humphrey <jasper521@gmail.com>

Based on: jmxetric by Jasper Humphrey (BSD style license)

License: BSD terms

   Copyright (C) 2010-2012 Daniel Pocock <daniel@pocock.com.au>
   Copyright (c) 2008-2011 Jasper Humphrey <jasper521@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
 OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
jBCrypt is subject to the following license:

/*
* Copyright (c) 2006 Damien Miller <djm@mindrot.org>
*
* Permission to use, copy, modify, and distribute this software for any
* purpose with or without fee is hereby granted, provided that the above
* copyright notice and this permission notice appear in all copies.
*
* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*/
Copyright (c) 2002-2012, the original author or authors.
All rights reserved.

http://www.opensource.org/licenses/bsd-license.php

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the following
conditions are met:

Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with
the distribution.

Neither the name of JLine nor the names of its contributors
may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO
EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,

OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2013-2016 ZURB, inc.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
BSD-style Licenses
==

The Apache TinkerPop project bundles the following components under the BSD License:

```
    jcabi-log (com.jcabi:jcabi-log:0.14 - http://www.jcabi.com/jcabi-log) - for details, see deps/jcabi-log
    jcabi-manifests 1.1 (com.jcabi:jcabi-manifests:1.1 - http://www.jcabi.com/jcabi-manifests) - for details, see deps/jcabi-manifests
    JLine (jline:jline:2.12 - https://github.com/jline/jline2) - for details, see deps/jline2
    Kryo (com.esotericsoftware:kryo-shaded:3.0.1
 - https://github.com/EsotericSoftware/kryo)
      - shaded in gremlin-shaded to org.apache.tinkerpop.shaded.kryo
      - for details, see deps/kryo
    minlog (com.esotericsoftware:minlog:1.3.0 - https://github.com/EsotericSoftware/minlog)
      - shaded in gremlin-shaded to org.apache.tinkerpop.shaded.minlog
      - for details, see deps/minlog
```

==
MIT Licenses
==

The Apache TinkerPop project bundles the following components under the MIT License:

```
    JCL 1.1.1 implemented over SLF4J (org.slf4j:jcl-over-slf4j:1.7.12 - http://www.slf4j.org) - for details, see deps/slf4j
    SLF4J API Module (org.slf4j:slf4j-api:1.7.12 - http://www.slf4j.org) - for details, see deps/slf4j
    SLF4J LOG4J-12 Binding (org.slf4j:slf4j-log4j12:1.7.12 - http://www.slf4j.org) - for details, see deps/slf4j
    Foundation stylesheet for CodeRay (http://foundation.zurb.com)
 - for details, see deps/foundation
    normalize.css 2.1.2 (http://necolas.github.io/normalize.css/) - for details, see deps/normalize
```

==
Other Licenses
==

The Apache TinkerPop project bundles the following components under the BSD/MIT License:

```
    gmetric4j (info.ganglia.gmetric4j:gmetric4j:1.0.3 - http://github.com/ganglia/gmetric4j) - for details, see deps/gmetric4j
```

The Apache TinkerPop project bundles the following components under the ISC License:

```
    jBCrypt (com.github.jeremyh:jBCrypt:0.4 - https://github.com/jeremyh/jBCrypt) - for details, see licenses/jbcrypt
```

Apache TinkerPop (incubating)

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache Commons Codec (AL ASF)
----
src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

Apache Commons Lang (AL ASF)

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

Apache Groovy (AL ASF)

This
product includes/uses ANTLR (http://www.antlr2.org/)
developed by Terence Parr 1989-2006

This product bundles icons from the famfamfam.com silk icons set
http://www.famfamfam.com/lab/icons/silk/
Licensed under the Creative Commons Attribution Licence v2.5
http://creativecommons.org/licenses/by/2.5/

Apache Http Components Core (AL ASF)

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

----
Apache Ivy (AL ASF)

Portions of Ivy were originally developed by
Jayasoft SARL (http://www.jayasoft.fr/)
and are licensed to the Apache Software Foundation under the
"Software Grant License Agreement"

SSH and SFTP
support is provided by the JCraft JSch package,
which is open source software, available under
the terms of a BSD style license.
The original software and related information is available
at http://www.jcraft.com/jsch/.

JavaTuples

Copyright (c) 2010, The JAVATUPLES team (http://www.javatuples.org)

Netty

Copyright 2014 The Netty Project

1.44 testify 1.1.3

1.44.1 Available under license :

Copyright (c) 2012 - 2013 Mat Ryer and Tyler Bunnell

Please consider promoting this project if you find it useful.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2013, Patrick Mezard
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2012-2013 Dave Collins <dave@davec.name>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
objx - by Mat Ryer and Tyler Bunnell

The MIT License (MIT)

Copyright (c) 2014 Stretchr, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.45 moment 2.19.3

1.45.1 Available under license :

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.46 react-redux 5.0.6

1.46.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

1.47 commons-compress 1.4.1

1.47.1 Available under license :

Apache Commons Compress
Copyright 2002-2012 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.48 kafka-node 1.6.2

1.48.1 Available under license :

1.49 antlr 3.5.2

1.49.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/Dbg.stg
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/Dbg.stg
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/serialize.g

No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations()
```

```
 ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _first_0 = null;
<ASTLabelType> _last = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
   retval.tree = (<ASTLabelType>)adaptor.getParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if
 ( state.backtracking==0 )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
>>

/**
 ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( state.backtracking==0 )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>
```

```
/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if
 ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0")
 ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>
```

```
matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.getTree());
<else>
 <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(), root_<treeLevel>);
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp()
 ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Perl5/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the

```
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
import org.antlr.runtime.tree.*;<\n>
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected var adaptor:TreeAdaptor = new CommonTreeAdaptor();<\n>
override public function set treeAdaptor(adaptor:TreeAdaptor):void {
   this.adaptor = adaptor;

 <grammar.directDelegates:{g|<g:delegateName()>.treeAdaptor = this.adaptor;}>
}
override public function get treeAdaptor():TreeAdaptor {
   return adaptor;
}
>>

@returnScope.ruleReturnMembers() ::= <<
<ASTLabelType> tree;
public function get tree():Object { return tree; }
>>

/** Add a variable to track rule's return AST */
```

```
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var root_0:<ASTLabelType> = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
 ruleDescriptor.wildcardTreeListLabels]:{it |var <it.label.text>_tree:<ASTLabelType>=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it |var <it.label.text>_tree:<ASTLabelType>=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{it |var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{it |var stream_<it>:RewriteRuleSubtreeStream=new
 RewriteRuleSubtreeStream(adaptor,"rule <it>");}; separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = <ASTLabelType>(adaptor.nil());<\n>
<endif>
<endif>
<endif>
>>

// T r a c k i n g  R u l e  E l e m e n t s

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions)
 ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
```

```
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(label, {<label>.tree})>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels];
 separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;
<rewriteCodeLabels()>
root_0 = <ASTLabelType>(adaptor.nil());
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = <ASTLabelType>(adaptor.rulePostProcessing(root_0));
input.replaceChildren(adaptor.getParent(retval.start),
                  adaptor.getChildIndex(retval.start),
                  adaptor.getChildIndex(_last),
                  retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{it |var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token
 <it>",<it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{it |var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
   separator="\n"
>
<referencedWildcardLabels
   :{it |var  stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"wildcard
<it>",<it>);};
   separator="\n"
>
<referencedWildcardListLabels
   :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"wildcard
<it>",list_<it>);};
   separator="\n"
>
<referencedRuleLabels
   :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.tree:null);};
   separator="\n"
>
<referencedRuleListLabels
   :{it |var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"token
<it>",list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note
 it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext}; separator="||"> ) {
   <alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el
 | stream_<el>.hasNext}; separator="||">) ) {
   throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
var root_<treeLevel>:<ASTLabelType> = <ASTLabelType>(adaptor.nil());
```

```
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/**
 Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>));<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>,
 <createImaginaryNode(token,terminalOptions,args)>);<\n>
>>
```

```
rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<createImaginaryNode(token,terminalOptions,args)>,
root_<treeLevel>));<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>));<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action)
 ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<action>, root_<treeLevel>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>
```

```
/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>



createImaginaryNode(tokenType,terminalOptions,args)
 ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
<ASTLabelType>(adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>))
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ActionScript/AST.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2006 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

```
/*
New style messages. This file contains the actual layout of the messages emitted by ANTLR.
The text itself is coming out of the languages/*stg files, according to the chosen locale.
This file contains the default format ANTLR uses.
*/

group antlr;

location(file, line, column) ::= "<file>:<line>:<column>:"

message(id, text) ::= "(<id>) <text>"

report(location, message, type) ::= "<type>(<message.id>): <location> <message.text>"

wantsSingleLineMessage() ::= "false"
```

Found in path(s):
*
 /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/messages/formats/antlr.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/AST.stg
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/ASTParser.stg
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ActionScript/ASTParser.stg
No license file was found, but licenses were detected in source scan.

IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/RubyTarget.java
No license file was found, but licenses were detected in source scan.

/* [The "BSD license"]
Copyright (c) 2008 Erik van Bilsen
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES,
 INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
group Delphi;

csharpTypeInitMap ::= [
 "int":"0",
 "uint":"0",
 "long":"0",
 "ulong":"0",
 "float":"0.0",
 "double":"0.0",
 "bool":"False",
 "byte":"0",
 "sbyte":"0",
 "short":"0",
 "ushort":"0",
 "char":"#0",
 "string":"""",
 "String":"""",
 default:"nil" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods
 for rules
*  and cyclic DFAs plus support code.
*  LEXER (Boolean): should we generate lexer code?
*  PARSER (Boolean): should we generate parser code?
*  TREE_PARSER (Boolean): should we generate tree parser code?
*  actionScope (String): 'lexer', 'parser', 'tree_parser' or custom scope
*  actions (HashMap):
*  docComment (String): document comment
*  recognizer (Object): recognizer class generator
*  name (String): name of grammar
*  tokens (HashMap<name: String, type: Integer>):
*  tokenNames:
*  rules:
*  cyclicDFAs:
*  bitsets:
*  buildTemplate (Boolean): should we generate a string template?
*  buildAST (Boolean): should we generate an AST?
*  rewriteMode (Boolean): are we rewriteing nodes?
*  profile (Boolean):
*  backtracking (Boolean): backtracking mode?
*  synpreds (): syntactic predicates
*  memoize (Boolean): should we memoize?
*  numRules (Integer): number of rules
*  fileName (String): fully qualified name of original .g file
*  ANTLRVersion
 (String): ANTLR version in Major.Minor.Build format
*  generatedTimestamp (String): date/time when the file is generated
*  trace (Boolean): should we trace input/output?
*  scopes:
*  superClass (String): name of base class, or empty string
*  literals:
*/
```

```
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
unit <name>;

{$HINTS OFF}

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<actions.(actionScope).header>

interface

<@imports>
uses<\n>
<@end>
 <actions.(actionScope).usesInterface>
<if(TREE_PARSER)>
 Antlr.Runtime.Tree,<\n>
<endif>
 Antlr.Runtime,
 Antlr.Runtime.Collections,
 Antlr.Runtime.Tools;

<docComment>
<recognizer>
>>

/** Generates source code for the
 lexer class
* grammar (Grammar object)
*/
lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="Lexer") ::= <<
type
 I<grammar.recognizerName> = interface(I<@superClassName><superClass><@end>)
 end;

 T<grammar.recognizerName> = class(T<@superClassName><superClass><@end>,
I<grammar.recognizerName>)
 strict private
   FCnt: array [0..<grammar.numberOfDecisions>] of Byte;
   FLA: array [0..<grammar.numberOfDecisions>, 0..255] of Integer;
   FException: ERecognitionException;
   procedure InitializeCyclicDFAs;
 <cyclicDFAs:cyclicDFADeclaration()>
 public
   const
     <tokens:{<it.name> = <it.type>;}; separator="\n">
   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 strict private
   <actions.(actionScope).memberDeclarations>
 public
   // delegates
   <grammar.delegates: {g|<g:delegateName()>: I<superClass>; {<g.recognizerName>}}; separator="\n">
 public
   // delegators
   <grammar.delegators:
 {g|<g:delegateName()>: Pointer; {<g.recognizerName>}}; separator="\n">
   <last(grammar.delegators):{g|gParent: Pointer; {<g.recognizerName>}}>
 protected
   { IBaseRecognizer }
   function GetGrammarFileName: String; override;
<if(filterMode)>
   function AlreadyParsedRule(const Input: IIntStream;
     const RuleIndex: Integer): Boolean; override;
   procedure Memoize(const Input: IIntStream; const RuleIndex,
     RuleStartIndex: Integer); override;
 protected
   { ILexer }
   function NextToken: IToken; override;<\n>
<endif>
 protected
   { ILexer }
   procedure DoTokens; override;
 public
   constructor Create; overload;
   constructor Create(const AInput: ICharStream<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>); overload;
   constructor Create(const AInput: ICharStream; const AState: IRecognizerSharedState<grammar.delegators:{g|;
const A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>); overload;

    <rules: {r | <if(!r.ruleDescriptor.isSynPred)><lexerRuleDeclaration(r)><endif>}>
   <synpreds:{p | <lexerSynpredDeclaration(p)>}; separator="\n">
 end;

implementation

uses
 <grammar.delegates: {g|<g.recognizerName>,}; separator="\n">
 <grammar.delegators: {g|<g.recognizerName>,}; separator="\n">
 <actions.(actionScope).usesImplementation>
 SysUtils,
 StrUtils,
 Math;

{ T<grammar.recognizerName> }

constructor T<grammar.recognizerName>.Create;
begin
 InitializeCyclicDFAs;
end;

constructor T<grammar.recognizerName>.Create(const AInput: ICharStream<grammar.delegators:{g|; const
A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>);
begin
 Create(AInput, nil<grammar.delegators:{g|, A<g:delegateName()>}>);
end;

constructor T<grammar.recognizerName>.Create(const AInput: ICharStream; const AState:
IRecognizerSharedState<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>);
begin
 inherited Create(AInput, AState);
 InitializeCyclicDFAs;
 { TODO: Necessary in Delphi??? Not removed yet. }
 <if(memoize)>
 <if(grammar.grammarIsRoot)>
 State.RuleMemoCount := <numRules>+1;<\n> <! index from 1..n !>
 <endif>
 <endif>
 <grammar.directDelegates:
  {g|<g:delegateName()> := T<g.recognizerName>.Create(AInput, State<trunc(g.delegators):{p|,
<p:delegateName()>}>, Self);}; separator="\n">
 <grammar.delegators:
  {g|<g:delegateName()> := Pointer(A<g:delegateName()>);}; separator="\n">
 <last(grammar.delegators):{g|gParent := Pointer(A<g:delegateName()>);}>
 <actions.(actionScope).memberInitializations>
end;
<actions.(actionScope).memberImplementations>
function T<grammar.recognizerName>.GetGrammarFileName: String;
begin
 Result := '<fileName>';
end;

<if(filterMode)>
<filteringNextToken()>
<endif>
```

```
<rules; separator="\n\n">
<synpreds:{p | <lexerSynpred(p)>}>

procedure T<grammar.recognizerName>.InitializeCyclicDFAs;
begin
 <cyclicDFAs:{dfa | FDFA<dfa.decisionNumber> :=
TDFA<dfa.decisionNumber>.Create(Self<@debugAddition()>);};
 separator="\n">
 <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>FDFA<dfa.decisionNumber>.SpecialStateTransitionHandler :=
DFA<dfa.decisionNumber>_SpecialStateTransition;<endif>}; separator="\n">
end;

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
end.>>

lexerRuleDeclaration(rule) ::= <<
procedure m<rule.ruleName>(<rule.ruleDescriptor.parameterScope:parameterScope(scope=rule)>);<\n>
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
function T<grammar.recognizerName>.NextToken: IToken;
var
 M: Integer;
begin
 while (True) do
 begin
   if (Input.LA(1) = Integer(cscEOF)) then
     Exit(TToken.EOF_TOKEN);

   State.Token := nil;
   State.Channel := TToken.DEFAULT_CHANNEL;
    State.TokenStartCharIndex := Input.Index;
   State.TokenStartCharPositionInLine := Input.CharPositionInLine;
   State.TokenStartLine := Input.Line;
   State.Text := '';
   try
     M := Input.Mark();
     State.Backtracking := 1; <! means we won't throw slow exception !>
     State.Failed := False;
     mTokens();
     State.Backtracking := 0;
```

```
<!
     mTokens backtracks with synpred at backtracking==2
           and we set the synpredgate to allow actions at level 1.
!>
     if (State.Failed) then
     begin
       Input.Rewind(M);
       Input.Consume; <! // advance one char and try again !>
     end
     else
     begin
       Emit;
       Exit(State.Token);
     end;
   except
     on RE: ERecognitionException do
     begin
       // shouldn't happen in backtracking mode, but...
       ReportError(RE);
       Recover(RE);
     end;
   end;
 end;
end;

function T<grammar.recognizerName>.AlreadyParsedRule(const Input: IIntStream;
 const
 RuleIndex: Integer): Boolean;
begin
 if (State.Backtracking > 1) then
   Result := inherited AlreadyParsedRule(Input, RuleIndex)
 else
   Result := False;
end;

procedure T<grammar.recognizerName>.Memoize(const Input: IIntStream; const RuleIndex,
 RuleStartIndex: Integer);
begin
 if (State.Backtracking > 1) then
   inherited Memoize(Input, RuleIndex, RuleStartIndex);
end;

>>

filteringActionGate() ::= "(State.Backtracking = 1)"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass, filterMode,
            ASTLabelType="ANTLRInterface", labelType, members, rewriteElementType) ::= <<
type
 <rules: {r | <genericParserRuleReturnType(rule=r, ruleDescriptor=r.ruleDescriptor)>}>
 I<grammar.recognizerName> = interface(I<@superClassName><superClass><@end>)
   <rules: {r | <genericParserRuleInterface(rule=r, ruleDescriptor=r.ruleDescriptor)>}>
 end;

 T<grammar.recognizerName>
 = class(T<@superClassName><superClass><@end>, I<grammar.recognizerName>)
<if(grammar.grammarIsRoot)>
 public
   const
     TOKEN_NAMES: array [0..<length(tokenNames)>+3] of String = (
       '\<invalid>',
       '\<EOR>',
       '\<DOWN>',
       '\<UP>',
       <tokenNames; separator=",\n">);<\n>
<endif>
 public
   const
     <tokens:{<it.name> = <it.type>;}; separator="\n">
 public
   // delegates
   <grammar.delegates: {g|<g:delegateName()>: I<superClass>; {<g.recognizerName>}}; separator="\n">
 public
   // delegators
   <grammar.delegators: {g|<g:delegateName()>: Pointer; {<g.recognizerName>}}; separator="\n">
   <last(grammar.delegators):{g|gParent: Pointer; {<g.recognizerName>}}>

   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeDeclaration(scope=it)><endif>}>
<@members>
   <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
 public
   constructor Create(const AInput: <inputStreamType><grammar.delegators:{g|; const A<g:delegateName()>:
 IBaseRecognizer{<g.recognizerName>}}>); overload;
   constructor Create(const AInput: <inputStreamType>; const AState:
IRecognizerSharedState<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>); overload;
<@end>
 protected
   { IBaseRecognizer }
   function GetTokenNames: TStringArray; override;
   function GetGrammarFileName: String; override;
 strict private
   <actions.(actionScope).memberDeclarations>
 <rules: {r | <genericParserRuleDeclaration(rule=r, ruleDescriptor=r.ruleDescriptor)>}>
```

```
<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
   // Delegated rules
   <grammar.delegatedRules:{ruleDescriptor| <delegatedRuleDeclaration(ruleDescriptor)>}>

   <synpreds:{p | <synpredDeclaration(p)>}; separator="\n">
 <cyclicDFAs:cyclicDFADeclaration()>
 strict private
   FException: ERecognitionException;
   FLA: array [0..<grammar.numberOfDecisions>, 0..255] of Integer;
   FCnt:
 array [0..<grammar.numberOfDecisions>] of Byte;
   procedure InitializeCyclicDFAs;
<if(bitsets)>
 public
   class var
     <bitsets:bitsetDecl(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>})>
 public
   class procedure InitializeBitsets; static;<\n>
<endif>
 end;

implementation

uses
 <grammar.delegates: {g|<g.recognizerName>,}; separator="\n">
 <grammar.delegators: {g|<g.recognizerName>,}; separator="\n">
 <actions.(actionScope).usesImplementation>
 SysUtils,
 StrUtils,
 Math;

{ T<grammar.recognizerName> }

constructor T<grammar.recognizerName>.Create(const AInput: <inputStreamType><grammar.delegators:{g|; const
A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>);
begin
 Create(AInput, TRecognizerSharedState.Create<grammar.delegators:{g|, A<g:delegateName()>}>);
end;

constructor T<grammar.recognizerName>.Create(const AInput: <inputStreamType>;
 const AState: IRecognizerSharedState<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>);
begin
  inherited Create(AInput, AState);
 <@membersConstructor>
 <@end>
 <parserCtorBody()>
 <grammar.directDelegates:{g|<g:delegateName()> := T<g.recognizerName>.Create(Input,
State<trunc(g.delegators):{p|, <p:delegateName()>}>, Self);}; separator="\n">
 <grammar.indirectDelegates:{g | <g:delegateName()> := <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
 <last(grammar.delegators):{g|gParent := Pointer(A<g:delegateName()>);}>
 <rules: {r | <ruleAttributeScopeInit(scope=r.ruleDescriptor.ruleScope)>}>
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <actions.(actionScope).memberInitializations>
end;
<actions.(actionScope).memberImplementations>

<grammar.delegatedRules:{ruleDescriptor| <delegatedRuleImplementation(ruleDescriptor)>}; separator="\n">
procedure T<grammar.recognizerName>.InitializeCyclicDFAs;
begin
 <cyclicDFAs:{dfa | FDFA<dfa.decisionNumber> := TDFA<dfa.decisionNumber>.Create(Self);}; separator="\n">
  <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>FDFA<dfa.decisionNumber>.SpecialStateTransitionHandler :=
DFA<dfa.decisionNumber>_SpecialStateTransition;<endif>}; separator="\n">
end;

<if(bitsets)>
class procedure T<grammar.recognizerName>.InitializeBitsets;
begin
 <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>
end;
<endif>

<@membersImplementation>
<@end>

function T<grammar.recognizerName>.GetTokenNames: TStringArray;
var
 I: Integer;
begin
 SetLength(Result,Length(T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES));
 for I := 0 to Length(T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES) - 1 do
   Result[I] := T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES[I];
end;

function T<grammar.recognizerName>.GetGrammarFileName: String;
begin
 Result := '<fileName>';
end;

<rules; separator="\n\n">
<synpreds:{p | <synpred(p)>}>

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
```

```
<if(bitsets)>
initialization
  T<grammar.recognizerName>.InitializeBitsets;<\n>
<endif>
end.>>

delegatedRuleDeclaration(ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<returnType()>;<\n>
<else>
procedure <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);<\n>
<endif>
<endif>
>>

delegatedRuleImplementation(ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
function
T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
): I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function
T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
): <returnType()>;<\n>
<else>
procedure
T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
);<\n>
<endif>
<endif>
begin
 <if(ruleDescriptor.hasReturnValue)>Result :=<endif>
T<ruleDescriptor.grammar.recognizerName>(<ruleDescriptor.grammar:delegateName()>.Implementor).<ruleDescri
ptor.name>(<ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", ">);
end;

>>

parserCtorBody() ::= <<
InitializeCyclicDFAs;
<if(memoize)>
<if(grammar.grammarIsRoot)>
```

```
State.RuleMemoCount := <length(grammar.allImportedRules)>+1;<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators: {g|<g:delegateName()> := Pointer(A<g:delegateName()>);}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType, superClass="Parser",
labelType="Token", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
 *  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start "<ruleName>"
procedure
T<grammar.recognizerName>.<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
;
var
 Alt: array [0..<grammar.numberOfDecisions>]
 of Integer;
 <ruleLabelDefVars()>
begin
 <ruleLabelDefs()>
<if(trace)>
 TraceIn('<ruleName>_fragment', <ruleDescriptor.index>);
 try
   <block>
 finally
   TraceOut('<ruleName>_fragment', <ruleDescriptor.index>);
 end;
<else>
 <block>
```

```
<endif>
end;
// $ANTLR end "<ruleName>"
>>

synpredDecls(name) ::= <<
SynPredPointer <name>;<\n>
>>

synpred(name) ::= <<

function T<grammar.recognizerName>.<name>: Boolean;
var
 Start: Integer;
 Success: Boolean;
begin
 State.Backtracking := State.Backtracking + 1;
 <@start()>
 Start := Input.Mark;
 try
   <name>_fragment(); // can never throw exception
 except
   on RE: ERecognitionException do
     WriteLn('Impossible: ' + RE.ToString);
 end;
 Success := not State.Failed;
 Input.Rewind(Start);
 <@stop()>
 State.Backtracking := State.Backtracking - 1;
 State.Failed := False;
 Result := Success;
end;<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

lexerSynpredDeclaration(name) ::= <<
function
 <name>: Boolean;
procedure <name>_fragment;
>>

synpredDeclaration(name) ::= <<
function <name>: Boolean;
procedure <name>_fragment;
>>
```

```
ruleMemoization(name) ::= <<
<if(memoize)>
if ((State.Backtracking > 0) and AlreadyParsedRule(Input, <ruleDescriptor.index>)) then
 Exit(<ruleReturnValue()>);
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)><\n>if (State.Failed) then Exit(<ruleReturnValue()>);<\n><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (State.Backtracking > 0) then
begin
 State.Failed := True;
 Exit(<ruleReturnValue()>);
end;<endif>
>>

genericParserRuleDeclaration(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.isSynPred)>
<else>
<ruleAttributeScopeDeclaration(scope=ruleDescriptor.ruleScope)>
<returnScopeDeclaration(scope=ruleDescriptor.returnScope)>
public
<if(ruleDescriptor.hasMultipleReturnValues)>
 function <rule.ruleName>:
 I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
 function <rule.ruleName>: <returnType()>;<\n>
<else>
 procedure <rule.ruleName>;<\n>
<endif>
<endif>
<endif>
>>

genericParserRuleInterface(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.isSynPred)>
<else>
<if(ruleDescriptor.hasMultipleReturnValues)>
function <rule.ruleName>: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
```

```
function <rule.ruleName>: <returnType()>;<\n>
<else>
procedure <rule.ruleName>;<\n>
<endif>
<endif>
<endif>
>>

genericParserRuleReturnType(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(ruleDescriptor.isSynPred)>
<else>
I<returnType()> = interface(I<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope)
end;<\n>
<endif>
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize)
 ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
(* <fileName>:<description> *)
<if(ruleDescriptor.hasMultipleReturnValues)>
function T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
I<returnType()>;
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<returnType()>;
<else>
procedure
T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);
<endif>
<endif>

var
<ruleDescriptor.actions.vars>
 Locals: TLocalStorage;
<if(ruleDescriptor.hasMultipleReturnValues)>
 RetVal: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
```

```
 RetVal: <returnType()>;<\n>
<else>
<endif>
<endif>
 Alt: array [0..<grammar.numberOfDecisions>] of Integer;
 <ruleDeclarationVars()>
 <ruleLabelDefVars()>
begin
 Locals.Initialize;
  try
   <if(trace)>TraceIn('<ruleName>', <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   try
     try
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
<if(exceptions)>
       <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
       <actions.(actionScope).rulecatch>
<else>
     except
       on RE: ERecognitionException do
       begin
         ReportError(RE);
         Recover(Input,RE);
         <@setErrorReturnValue()>
       end;<\n>
<endif>
<endif>
<endif>
     end;
   finally
     <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
     <memoize()>
     <ruleScopeCleanUp()>
     <finally>
   end;
   <@postamble()>
 finally
```

```
   Locals.Finalize;
 end;
 Exit(<ruleReturnValue()>);
end;
//
 $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
   <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal := T<returnType()>.Create;
RetVal.Start := Input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name> := <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex := Input.Index();
<endif>
>>

ruleDeclarationVars() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name>: <a.type>;
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex: Integer;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>Stack.Push(T<it>Scope.Create);}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>Stack.Push(T<it.name>Scope.Create);}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>Stack.Pop();}; separator="\n">
```

```
<ruleDescriptor.ruleScope:{<it.name>Stack.Pop;};
 separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]:{<it.label.text> := nil;}; separator="\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{list_<it.label.text> := nil;}; separator="\n">
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll.label.text> := nil;}; separator="\n">
>>

ruleLabelDefVars() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]:{<it.label.text>: I<labelType>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{list_<it.label.text>: IList\<IANTLRInterface\>;};
separator="\n">
<ruleDescriptor.ruleLabels:ruleLabelDefVar(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll.label.text>: <ruleLabelType(referencedRule=ll.referencedRule)>;};
separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
    :{<it.label.text> := nil;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{List_<it.label.text> := nil;}; separator="\n"
>
>>

lexerRuleLabelDefDeclarations() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<it.label.text>: I<labelType>;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{List_<it.label.text>: IList;}; separator="\n"
>
>>
```

```
ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
RetVal
<endif>
<else>
<! nil !>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
RetVal.Stop
 := Input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (State.Backtracking > 0) then
 Memoize(Input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex);
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
<ruleDescriptor.parameterScope>
procedure
T<grammar.recognizerName>.m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);
var
 <ruleDescriptor.actions.vars>
 Locals: TLocalStorage;
 TokenType, Channel: Integer;
 Alt: array [0..<grammar.numberOfDecisions>] of Integer;
 <lexerRuleLabelDefDeclarations()>
begin
 Locals.Initialize;
 try
   <ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
   <if(trace)>TraceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   try
<if(nakedBlock)>

   <ruleMemoization(name=ruleName)>
     <lexerRuleLabelDefs()>
     <ruleDescriptor.actions.init>
     <block><\n>
<else>
     TokenType := <ruleName>;
     Channel := DEFAULT_TOKEN_CHANNEL;
     <ruleMemoization(name=ruleName)>
     <lexerRuleLabelDefs()>
     <ruleDescriptor.actions.init>
     <block>
     <ruleCleanUp()>
     State.TokenType := TokenType;
     State.Channel := Channel;
     <(ruleDescriptor.actions.after):execAction()>
<endif>
   finally
     <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
     <ruleScopeCleanUp()>
     <memoize()>
   end;
 finally
   Locals.Finalize;
 end;
end;
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
procedure T<grammar.recognizerName>.mTokens;
var
 Alt: array [0..<grammar.numberOfDecisions>] of Integer;
begin
 <block>
end;

procedure T<grammar.recognizerName>.DoTokens;
```

```
begin
  mTokens;
end;
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
Alt[<decisionNumber>] := <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
case Alt[<decisionNumber>] of
 <alts:altSwitchCase()>
end;
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
Alt[<decisionNumber>] := <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
case Alt[<decisionNumber>] of
 <alts:altSwitchCase()>
end;
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
(* <fileName>:<description> *)
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule
 with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
(* <fileName>:<description> *)
<decls>
```

```
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
(* <fileName>:<description> *)
FCnt[<decisionNumber>] := 0;
<decls>
<@preloop()>
while (True) do
begin
 Alt[<decisionNumber>] := <maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 case Alt[<decisionNumber>] of
   <alts:altSwitchCase()>
 else
   begin
     if (FCnt[<decisionNumber>] >= 1) then
       Break;
     <ruleBacktrackFailure()>
     raise EEarlyExitException.Create(<decisionNumber>, Input);
     <@earlyExitException()>
   end;
 end;
 Inc(FCnt[<decisionNumber>]);
end;
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<
(* <fileName>:<description> *)
<decls>
<@preloop()>
while (True) do
begin
 Alt[<decisionNumber>] := <maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 case Alt[<decisionNumber>] of
   <alts:altSwitchCase()>
 else
   Break;
 end;
end;
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
<i>:
 <@prealt()>
 <it><\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew)
 ::= <<
(* <fileName>:<description> *)
begin
 <@declarations()>
 <elements:element()>
 <rew>
 <@cleanup()>
end;
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
```

```
element() ::= <<
<@prematch()>
<it.el>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label> := <endif>Match(Input, <token>,
FOLLOW_<token>_in_<ruleName><elementIndex>)<if(label)> as
I<labelType><endif>;<\n><checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> = nil) then list_<label> := TList\<IANTLRInterface\>.Create;
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
<endif>
Match(<char>);
 <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
<endif>
MatchRange(<a>, <b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label> := Input.LA(1);<\n>
<else>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
```

```
<endif>
if (<s>) then
begin
 Input.Consume;
 <postmatchCode>
 <if(!LEXER)>
 State.ErrorRecovery := False;<endif>
 <if(backtracking)>State.Failed := False;<endif>
end
else
begin
 <ruleBacktrackFailure()>
 FException := EMismatchedSetException.Create(nil, Input);
 <@mismatchedSetException()>
<if(LEXER)>
 Recover(FException);
 raise FException;<\n>
<else>
 raise FException;
 <! use following code to make it recover inline; remove throw mse;
 RecoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
  !>
<endif>
end;<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
Locals.AsInteger['<label>Start'] := CharIndex;
Match(<string>); <checkRuleBacktrackFailure()>
<label> := TCommonToken.Create(Input, TToken.INVALID_TOKEN_TYPE, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start'], CharIndex-1);
<else>
Match(<string>); <checkRuleBacktrackFailure()>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label> := Input.LT(1) as I<labelType>;<\n>
```

```
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label,
 elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
<label> := <if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
<else>
<if(scope)>T<scope.recognizerName>(IANTLRObject(<scope:delegateName()>).Implementor).<endif><rule.name
>(<args; separator=", ">);<\n>
<endif>
State.FollowingStackPointer := State.FollowingStackPointer - 1;
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
```

```
*
*  The 'rule' argument was
 the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
Locals.AsInteger['<label>Start<elementIndex>'] := CharIndex;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> := TCommonToken.Create(Input, TToken.INVALID_TOKEN_TYPE, TToken.DEFAULT_CHANNEL,
 Locals.AsInteger['<label>Start<elementIndex>'], CharIndex - 1);
<else>
<if(scope)>(<scope:delegateName()>.Implementor as T<scope.recognizerName>).<endif>m<rule.name>(<args;
separator=", ">); <checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
Locals.AsInteger['<label>Start<elementIndex>'] := CharIndex;
Match(EOF);
 <checkRuleBacktrackFailure()>
Locals['<label>'] := TCommonToken.Create(Input, EOF, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start<elementIndex>'], CharIndex-1);
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (Input.LA(1) = TToken.DOWN) then
begin
 Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
end;
<else>
Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
<children:element()>
Match(Input, TToken.UP, nil);<\n><checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if (not (<evalPredicate(...)>))
 then
begin
 <ruleBacktrackFailure()>
 raise EFailedPredicateException.Create(Input, '<ruleName>', '<description>');
end;<\n>
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
begin
<if(eotPredictsAlt)>
 Alt[<decisionNumber>] := <eotPredictsAlt>;<\n>
<else>
 <ruleBacktrackFailure()>
 raise ENoViableAltException.Create('<description>', <decisionNumber>, <stateNumber>, Input);<\n>
<endif>
end;
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);<\n>
<edges;
 separator="\nelse ">;
>>
```

```
/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);
<edges; separator="\nelse ">;<\n>
<if(eotPredictsAlt)>
<if(!edges)>
Alt[<decisionNumber>] := <eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else
begin
 Alt[<decisionNumber>] := <eotPredictsAlt>;
end;<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "Alt[<decisionNumber>] := <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions,
 we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> and (<predicates>)<endif>) then
begin
 <targetState>
end <! no ; here !>
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
 <edges; separator="\n">
else
 begin
<if(eotPredictsAlt)>
   Alt[<decisionNumber>] := <eotPredictsAlt>;
```

```
<else>
   <ruleBacktrackFailure()>
   <@noViableAltException()>
   raise ENoViableAltException.Create('<description>', <decisionNumber>, <stateNumber>, Input);<\n>
<endif>
 end;
end;<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
 <edges; separator="\n">
end;<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
case Input.LA(<k>) of
 <edges; separator="\n"><\n>
<if(eotPredictsAlt)>
else
 Alt[<decisionNumber>] := <eotPredictsAlt>;<\n>
<endif>
end;<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{<it>}; separator=",\n">:
 begin
   <targetState>
 end;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
Alt[<decisionNumber>] := FDFA<decisionNumber>.Predict(Input);
>>

/* Dump DFA tables.
*/
cyclicDFADeclaration(dfa) ::= <<
strict protected
 type
```

```
   TDFA<dfa.decisionNumber> = class(TDFA)
   protected
     { IDFA }
     function Description: String; override;
   public
     constructor Create(const ARecognizer: IBaseRecognizer);
   end;
 var
   FDFA<dfa.decisionNumber>: IDFA;
<if(dfa.specialStateSTs)>
strict
 protected
 function DFA<dfa.decisionNumber>_SpecialStateTransition(const DFA: IDFA; S: Integer;
   const AInput: IIntStream): Integer;<endif>
>>

cyclicDFA(dfa) ::= <<
{ T<grammar.recognizerName>.TDFA<dfa.decisionNumber> }

constructor T<grammar.recognizerName>.TDFA<dfa.decisionNumber>.Create(const ARecognizer:
IBaseRecognizer);
const
 DFA<dfa.decisionNumber>_EOT = '<dfa.javaCompressedEOT; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_EOF = '<dfa.javaCompressedEOF; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_MIN = '<dfa.javaCompressedMin; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_MAX = '<dfa.javaCompressedMax; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_ACCEPT = '<dfa.javaCompressedAccept; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_SPECIAL = '<dfa.javaCompressedSpecial; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_TRANSITION: array [0..<length(dfa.javaCompressedTransition)>-1] of String = (
   <dfa.javaCompressedTransition:{s|'<s; wrap="'+\n'">'}; separator=",\n">);
begin
  inherited Create;
 Recognizer := ARecognizer;
 DecisionNumber := <dfa.decisionNumber>;
 EOT := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_EOT);
 EOF := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_EOF);
 Min := TDFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_MIN);
 Max := TDFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_MAX);
 Accept := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_ACCEPT);
 Special := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_SPECIAL);
 Transition := TDFA.UnpackEncodedStringArray(DFA<dfa.decisionNumber>_TRANSITION);
end;

function T<grammar.recognizerName>.TDFA<dfa.decisionNumber>.Description: String;
begin
 Result := '<dfa.description>';
end;<\n>
<if(dfa.specialStateSTs)>
```

```
function T<grammar.recognizerName>.DFA<dfa.decisionNumber>_SpecialStateTransition(const DFA: IDFA; S:
Integer;
 const AInput: IIntStream): Integer;
var
 Locals: TLocalStorage;
 <if(LEXER)>
 Input: IIntStream;
 <endif>
  <if(PARSER)>
 Input: ITokenStream;
 <endif>
 <if(TREE_PARSER)>
 Input: ITreeNodeStream;
 <endif>
 _S: Integer;
 NVAE: ENoViableAltException;
begin
 Result := -1;
 Locals.Initialize;
 try
   <if(LEXER)>
   Input := AInput;
   <endif>
   <if(PARSER)>
   Input := AInput as ITokenStream;
   <endif>
   <if(TREE_PARSER)>
   Input := AInput as ITreeNodeStream;
   <endif>
   _S := S;
   case S of
     <dfa.specialStateSTs:{state | <i0>: begin<! compressed special state numbers 0..n-1 !>
    <state> <\n>   end;}; separator="\n">
   end;
<if(backtracking)>
   if (State.Backtracking > 0) then
   begin
     State.Failed := True;
     Exit(-1);
   end;<\n>
<endif>
   NVAE := ENoViableAltException.Create(DFA.Description, <dfa.decisionNumber>, _S, Input);
   DFA.Error(NVAE);
   raise NVAE;
 finally
   Locals.Finalize;
 end;
end;<\n>
```

```
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState)
 ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
Locals.AsInteger['index<decisionNumber>_<stateNumber>'] := Input.Index;
Input.Rewind;<\n>
<endif>
S := -1;
<edges; separator="\nelse ">;
<if(semPredState)> <! return input cursor to state before we rewound !>
Input.Seek(Locals.AsInteger['index<decisionNumber>_<stateNumber>']);<\n>
<endif>
if (S >= 0) then
 Exit(S);
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> and (<predicates>)<endif>) then
 S := <targetStateNumber>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
S := <targetStateNumber>;<\n>
>>



//
 D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "(<operands:{o|(<o>)}; separator=\" or \">)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"
```

```
evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "FLA[<decisionNumber>,<stateNumber>] = <atomAsInt>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "Input.LA(<k>) = <atomAsInt>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
((FLA[<decisionNumber>,<stateNumber>] \>= <lowerAsInt>) and (FLA[<decisionNumber>,<stateNumber>] \<=
<upperAsInt>))
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(Input.LA(<k>) \>=
<lowerAsInt>) and (Input.LA(<k>) \<= <upperAsInt>)"

setTest(ranges) ::= "<ranges; separator=\")
 or (\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<scope.name>Stack := TStackList\<I<scope.name>Scope\>.Create;<\n>
<endif>
>>

globalAttributeScopeDeclaration(scope) ::= <<
<if(scope.attributes)>
strict protected
 type
   I<scope.name>Scope = interface(IANTLRObject)
   end;
   T<scope.name>Scope = class(TANTLRObject, I<scope.name>Scope)
   protected
     <scope.attributes:{<it.name>: <it.type>;}; separator="\n">
   end;
strict protected
 <scope.name>Stack: IStackList\<I<scope.name>Scope\>;
<endif>
>>

ruleAttributeScopeDeclaration(scope) ::= <<
<if(scope.attributes)>
strict protected
 type
   I<scope.name>Scope = interface(IANTLRObject)
```

```
  end;
  T<scope.name>Scope = class(TANTLRObject, I<scope.name>Scope)
  protected
    <scope.attributes:{<it.name>: <it.type>;}; separator="\n">
  end;
strict protected
 <scope.name>Stack: IStackList\<I<scope.name>Scope\>;
<endif>
>>

ruleAttributeScope(scope) ::= <<
<! protected Stack <scope.name>Stack = new
 Stack();<\n> !>
>>

ruleAttributeScopeInit(scope) ::= <<
<if(scope)>
<scope.name>Stack := TStackList\<I<scope.name>Scope\>.Create;<\n>
<endif>
>>

returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()>
<! I<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope !>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
<! Pointer/void !>
<endif>
<endif>
>>

/** Generate the C# type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
I<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
```

```
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
 */
initValue(typeName) ::= <<
<csharpTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> := <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

ruleLabelDefVar(label) ::= <<
<label.label.text>: <ruleLabelType(referencedRule=label.referencedRule)>;
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
{ T<ruleDescriptor:returnStructName()> }

<scope.attributes:{public <it.decl>;}; separator="\n">
<@ruleReturnMembers()>
<endif>
>>

returnScopeDeclaration(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public
 type
   T<ruleDescriptor:returnStructName()> =
class(T<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope, I<ruleDescriptor:returnStructName()>)
   <scope.attributes:{public
 <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
   end;
<endif>
>>
```

```
parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> := <expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
(<scope>Stack[<scope>Stack.Count-<negIndex>-1] as T<scope>Scope).<attr.name>
<else>
<if(index)>
(<scope>Stack[<index>] as T<scope>Scope).<attr.name>
((<scope>_scope)<scope>_stack[<index>]).<attr.name>
<else>
(<scope>Stack.Peek.Implementor as T<scope>Scope).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
(<scope>Stack[<scope>Stack.Count-<negIndex>-1] as T<scope>Scope).<attr.name> := <expr>;<\n>
<else>
<if(index)>
(<scope>Stack[<index>] as T<scope>Scope).<attr.name> := <expr>;<\n>
<else>
(<scope>Stack.Peek.Implementor as T<scope>Scope).<attr.name> := <expr>;<\n>
<endif>
<endif>
>>

/** $x is
 either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>Stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(IfThen(Assigned(<scope>),Def(<scope>).<attr.name>,<initValue(attr.type)>))
<else>
<scope>
<endif>
>>
```

```
returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal.<attr.name> := <expr>;
<else>
<attr.name> := <expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next
 are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(Def(<scope>).Text)"
tokenLabelPropertyRef_type(scope,attr) ::= "(Def(<scope>).TokenType)"
tokenLabelPropertyRef_line(scope,attr) ::= "(Def(<scope>).Line)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(Def(<scope>).CharPositionInLine)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(Def(<scope>).Channel)"
tokenLabelPropertyRef_index(scope,attr) ::= "(Def(<scope>).TokenIndex)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(StrToIntDef(Def(<scope>).Text,0))"

ruleLabelPropertyRef_start(scope,attr) ::= "(IfThen(Assigned(<scope>), Def(<scope>).Start, nil) as I<labelType>)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(Def(<scope>).Stop as I<labelType>)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(Def(Def(<scope>).Tree as I<ASTLabelType>))"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
IfThen(Assigned(<scope>), Input.TokenStream.ToString(
 Input.TreeAdaptor.GetTokenStartIndex(Def(<scope>).Start),
  Input.TreeAdaptor.GetTokenStopIndex(Def(<scope>).Start)), '')
<else>
IfThen(Assigned(<scope>), Input.ToString(
 (Def(<scope>).Start) as IToken,(Def(<scope>).Stop) as IToken), '')
<endif>
```

```
>>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> != null) ? <scope>.ST : null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(Def(<scope>).TokenType)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(Def(<scope>).Line)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(IfThen(Assigned(<scope>),Def(<scope>).CharPositionInLine,-1))"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(Def(<scope>).Channel)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(Def(<scope>).TokenIndex)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(Def(<scope>).Text)"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "(StrToIntDef(Def(<scope>).Text,0))"

// Somebody may ref $template
 or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "(RetVal.Start as I<labelType>)"
rulePropertyRef_stop(scope,attr) ::= "(RetVal.Stop as I<labelType>)"
rulePropertyRef_tree(scope,attr) ::= "(RetVal.Tree as I<ASTLabelType>)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
Input.TokenStream.ToString(
 Input.TreeAdaptor.GetTokenStartIndex(RetVal.Start),
 Input.TreeAdaptor.GetTokenStopIndex(RetVal.Start))
<else>
Input.ToString(RetVal.Start as IToken,Input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "RetVal.ST"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "TokenType"
lexerRulePropertyRef_line(scope,attr) ::= "State.TokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "State.TokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "Channel"
lexerRulePropertyRef_start(scope,attr) ::= "State.TokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr)
 ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "StrToInt(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "RetVal.Tree := <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "RetVal.ST := <expr>;"


/** How to execute an action (only when not backtracking) */
```

```
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if (<actions.(actionScope).synpredgate>) then
begin
 <action>
end;
<else>
if (State.Backtracking = 0) then
begin
 <action>
end;<\n>
<endif>
<else>
<action>
<endif>
>>


/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
<name> := TBitSet.Create([<words64:{<it>};separator=",">]);<\n>
>>

bitsetDecl(name) ::= <<
<name>: IBitSet;<\n>
>>

codeFileExtension() ::= ".pas"

true()
 ::= "True"
false() ::= "False"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Delphi/Delphi.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2010 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
/*
New style messages. This file only contains the messages in English, but no
information about which file, line, or column it occurred in.
The location and message ids are taken out of the formats directory.
		Kay Roepke
*/
group en_US;

// TOOL ERRORS
// file errors
CANNOT_WRITE_FILE(arg,exception,stackTrace) ::= <<
cannot write file <arg>: <exception>
<stackTrace; separator="\n">
>>
CANNOT_CLOSE_FILE(arg,exception,stackTrace) ::= "cannot close file <arg>: <exception>"
CANNOT_FIND_TOKENS_FILE(arg)
 ::= "cannot find tokens file <arg>"
ERROR_READING_TOKENS_FILE(arg,exception,stackTrace) ::= <<
problem reading token vocabulary file <arg>: <exception>
<stackTrace; separator="\n">
>>
DIR_NOT_FOUND(arg,exception,stackTrace) ::= "directory not found: <arg>"
OUTPUT_DIR_IS_FILE(arg,exception,stackTrace) ::= "output directory is a file: <arg>"
CANNOT_OPEN_FILE(arg,exception,stackTrace) ::= "cannot find or open file: <arg><if(exception)>; reason: <exception><endif>"
CIRCULAR_DEPENDENCY() ::= "your grammars contain a circular dependency and cannot be sorted into a valid
```

```
build order."

INTERNAL_ERROR(arg,arg2,exception,stackTrace) ::= <<
internal error: <arg> <arg2><if(exception)>: <exception><endif>
<stackTrace; separator="\n">
>>
INTERNAL_WARNING(arg) ::= "internal warning: <arg>"
ERROR_CREATING_ARTIFICIAL_RULE(arg,exception,stackTrace) ::= <<
problems creating lexer rule listing all tokens: <exception>
<stackTrace; separator="\n">
>>
TOKENS_FILE_SYNTAX_ERROR(arg,arg2)
 ::=
 "problems parsing token vocabulary file <arg> on line <arg2>"
CANNOT_GEN_DOT_FILE(arg,exception,stackTrace) ::=
 "cannot write DFA DOT file <arg>: <exception>"
BAD_ACTION_AST_STRUCTURE(exception,stackTrace) ::=
 "bad internal tree structure for action '<arg>': <exception>"
BAD_AST_STRUCTURE(arg,exception,stackTrace) ::= <<
bad internal tree structure '<arg>': <exception>
<stackTrace; separator="\n">
>>
FILE_AND_GRAMMAR_NAME_DIFFER(arg,arg2) ::=
 "file <arg2> contains grammar <arg>; names must be identical"
FILENAME_EXTENSION_ERROR(arg) ::=
 "file <arg> must end in a file extension, normally .g"

// code gen errors
MISSING_CODE_GEN_TEMPLATES(arg) ::=
 "cannot find code generation templates <arg>.stg"
MISSING_CYCLIC_DFA_CODE_GEN_TEMPLATES() ::=
 "cannot find code generation cyclic DFA templates for language <arg>"
CODE_GEN_TEMPLATES_INCOMPLETE(arg) ::=
 "at least one code generation template missing for language <arg>"
CANNOT_CREATE_TARGET_GENERATOR(arg,exception,stackTrace)
 ::=
 "cannot create target <arg> code generator: <exception>"
STRING_TEMPLATE_ERROR(arg,exception,stackTrace) ::=
 "template error: <arg>"
CANNOT_COMPUTE_SAMPLE_INPUT_SEQ() ::=
 "cannot generate a sample input sequence from lookahead DFA"

// grammar interpretation errors
/*
NO_VIABLE_DFA_ALT(arg,arg2) ::=
 "no viable transition from state <arg> on <arg2> while interpreting DFA"
*/
```

```
// GRAMMAR ERRORS
SYNTAX_ERROR(arg) ::= "syntax error: <arg>"
RULE_REDEFINITION(arg) ::=
 "rule <arg> redefinition"
LEXER_RULES_NOT_ALLOWED(arg) ::=
 "lexer rule <arg> not allowed in parser"
PARSER_RULES_NOT_ALLOWED(arg) ::=
 "parser rule <arg> not allowed in lexer"
CANNOT_FIND_ATTRIBUTE_NAME_IN_DECL(arg) ::=
 "cannot find an attribute name in attribute declaration"
NO_TOKEN_DEFINITION(arg) ::=
 "no lexer rule corresponding to token: <arg>"
UNDEFINED_RULE_REF(arg) ::=
 "reference to undefined rule: <arg>"
LITERAL_NOT_ASSOCIATED_WITH_LEXER_RULE(arg) ::=
 "literal has no associated lexer rule: <arg>"
CANNOT_ALIAS_TOKENS_IN_LEXER(arg)
 ::=
 "literals are illegal in lexer tokens{} section: <arg>"
ATTRIBUTE_REF_NOT_IN_RULE(arg,arg2) ::=
 "reference to attribute outside of a rule: <arg><if(arg2)>.<arg2><endif>"
UNKNOWN_ATTRIBUTE_IN_SCOPE(arg,arg2) ::=
 "unknown attribute for <arg>: <arg2>"
UNKNOWN_RULE_ATTRIBUTE(arg,arg2) ::=
 "unknown attribute for rule <arg>: <arg2>"
UNKNOWN_SIMPLE_ATTRIBUTE(arg,args2) ::=
 "attribute is not a token, parameter, or return value: <arg>"
ISOLATED_RULE_SCOPE(arg) ::=
 "missing attribute access on rule scope: <arg>"
INVALID_RULE_PARAMETER_REF(arg,arg2) ::=
 "cannot access rule <arg>'s parameter: <arg2>"
INVALID_RULE_SCOPE_ATTRIBUTE_REF(arg,arg2) ::=
 "cannot access rule <arg>'s dynamically-scoped attribute: <arg2>"
SYMBOL_CONFLICTS_WITH_GLOBAL_SCOPE(arg) ::=
 "symbol <arg> conflicts with global dynamic scope with same name"
WRITE_TO_READONLY_ATTR(arg,arg2,arg3) ::=
 "cannot write to read only attribute: $<arg><if(arg2)>.<arg2><endif>"
LABEL_CONFLICTS_WITH_RULE(arg)
 ::=
 "label <arg> conflicts with rule with same name"
LABEL_CONFLICTS_WITH_TOKEN(arg) ::=
 "label <arg> conflicts with token with same name"
LABEL_CONFLICTS_WITH_RULE_SCOPE_ATTRIBUTE(arg,arg2) ::=
 "label <arg> conflicts with rule <arg2>'s dynamically-scoped attribute with same name"
LABEL_CONFLICTS_WITH_RULE_ARG_RETVAL(arg,arg2) ::=
 "label <arg> conflicts with rule <arg2>'s return value or parameter with same name"
ATTRIBUTE_CONFLICTS_WITH_RULE(arg,arg2) ::=
 "rule <arg2>'s dynamically-scoped attribute <arg> conflicts with the rule name"
```

ATTRIBUTE_CONFLICTS_WITH_RULE_ARG_RETVAL(arg,arg2) ::=
"rule <arg2>'s dynamically-scoped attribute <arg> conflicts with<arg2>'s return value or parameter with same name"
LABEL_TYPE_CONFLICT(arg,arg2) ::=
"label <arg> type mismatch with previous definition: <arg2>"
ARG_RETVAL_CONFLICT(arg,arg2) ::=
"rule <arg2>'s argument <arg> conflicts a return value with same name"
NONUNIQUE_REF(arg) ::=
"<arg> is a non-unique reference"
FORWARD_ELEMENT_REF(arg)
::=
"illegal forward reference: <arg>"
MISSING_RULE_ARGS(arg) ::=
"missing parameter(s) on rule reference: <arg>"
RULE_HAS_NO_ARGS(arg) ::=
"rule <arg> has no defined parameters"
ARGS_ON_TOKEN_REF(arg) ::=
"token reference <arg> may not have parameters"
ILLEGAL_OPTION(arg) ::=
"illegal option <arg>"
LIST_LABEL_INVALID_UNLESS_RETVAL_STRUCT(arg) ::=
"rule '+=' list labels are not allowed w/o output option: <arg>"
UNDEFINED_TOKEN_REF_IN_REWRITE(arg) ::=
"reference to undefined token in rewrite rule: <arg>"
REWRITE_ELEMENT_NOT_PRESENT_ON_LHS(arg) ::=
"reference to rewrite element <arg> without reference on left of ->"
UNDEFINED_LABEL_REF_IN_REWRITE(arg) ::=
"reference to undefined label in rewrite rule: $<arg>"
NO_GRAMMAR_START_RULE (arg) ::=
"grammar <arg>: no start rule (no rule can obviously be followed by EOF)"
EMPTY_COMPLEMENT(arg) ::= <<
<if(arg)>
set complement ~<arg> is empty
<else>
set complement is empty
<endif>
>>
UNKNOWN_DYNAMIC_SCOPE(arg) ::=
"unknown
dynamic scope: <arg>"
UNKNOWN_DYNAMIC_SCOPE_ATTRIBUTE(arg,arg2) ::=
"unknown dynamically-scoped attribute for scope <arg>: <arg2>"
RULE_REF_AMBIG_WITH_RULE_IN_ALT(arg) ::=
"reference $<arg> is ambiguous; rule <arg> is enclosing rule and referenced in the production (assuming enclosing rule)"
ISOLATED_RULE_ATTRIBUTE(arg) ::=
"reference to locally-defined rule scope attribute without rule name: <arg>"
INVALID_ACTION_SCOPE(arg,arg2) ::=

"unknown or invalid action scope for <arg2> grammar: <arg>"
ACTION_REDEFINITION(arg) ::=
"redefinition of <arg> action"
DOUBLE_QUOTES_ILLEGAL(arg) ::=
"string literals must use single quotes (such as \'begin\'): <arg>"
INVALID_TEMPLATE_ACTION(arg) ::=
"invalid StringTemplate % shorthand syntax: '<arg>'"
MISSING_ATTRIBUTE_NAME() ::=
"missing attribute name on $ reference"
ARG_INIT_VALUES_ILLEGAL(arg) ::=
"rule parameters may not have init values: <arg>"
REWRITE_OR_OP_WITH_NO_OUTPUT_OPTION(arg) ::=
"<if(arg)>rule <arg> uses <endif>rewrite
syntax or operator with no output option; setting output=AST"
AST_OP_WITH_NON_AST_OUTPUT_OPTION(arg) ::=
"AST operator with non-AST output option: <arg>"
NO_RULES(arg) ::= "grammar file <arg> has no rules"
MISSING_AST_TYPE_IN_TREE_GRAMMAR(arg) ::=
"tree grammar <arg> has no ASTLabelType option"
REWRITE_FOR_MULTI_ELEMENT_ALT(arg) ::=
"with rewrite=true, alt <arg> not simple node or obvious tree element; text attribute for rule not guaranteed to be correct"
RULE_INVALID_SET(arg) ::=
"Cannot complement rule <arg>; not a simple set or element"
HETERO_ILLEGAL_IN_REWRITE_ALT(arg) ::=
"alts with rewrites can't use heterogeneous types left of ->"
NO_SUCH_GRAMMAR_SCOPE(arg,arg2) ::=
"reference to undefined grammar in rule reference: <arg>.<arg2>"
NO_SUCH_RULE_IN_SCOPE(arg,arg2) ::=
"rule <arg2> is not defined in grammar <arg>"
TOKEN_ALIAS_CONFLICT(arg,arg2) ::=
"cannot alias <arg>; string already assigned to <arg2>"
TOKEN_ALIAS_REASSIGNMENT(arg,arg2) ::=
"cannot alias <arg>;
token name already assigned to <arg2>"
TOKEN_VOCAB_IN_DELEGATE(arg,arg2) ::=
"tokenVocab option ignored in imported grammar <arg>"
INVALID_IMPORT(arg,arg2) ::=
"<arg.grammarTypeString> grammar <arg.name> cannot import <arg2.grammarTypeString> grammar <arg2.name>"
IMPORTED_TOKENS_RULE_EMPTY(arg,arg2) ::=
"no lexer rules contributed to <arg> from imported grammar <arg2>"
IMPORT_NAME_CLASH(arg,arg2) ::=
"combined grammar <arg.name> and imported <arg2.grammarTypeString> grammar <arg2.name> both generate <arg2.recognizerName>; import ignored"
AST_OP_IN_ALT_WITH_REWRITE(arg,arg2) ::=
"rule <arg> alt <arg2> uses rewrite syntax and also an AST operator"
WILDCARD_AS_ROOT(arg) ::= "Wildcard invalid as root; wildcard can itself be a tree"

```
CONFLICTING_OPTION_IN_TREE_FILTER(arg,arg2) ::= "option <arg>=<arg2> conflicts with tree grammar filter mode"
ILLEGAL_OPTION_VALUE(arg, arg2) ::= "value '<arg2>' invalid for option <arg>"
ALL_OPS_NEED_SAME_ASSOC(arg) ::= "all operators of alt <alt> of
 left-recursive rule must have same associativity"
RANGE_OP_ILLEGAL(arg) ::= "the .. range operator isn't allowed in parser rules"

// GRAMMAR WARNINGS

GRAMMAR_NONDETERMINISM(input,conflictingAlts,paths,disabled,hasPredicateBlockedByAction) ::=
<<
<if(paths)>
Decision can match input such as "<input>" using multiple alternatives:
<paths:{ it |  alt <it.alt> via NFA path <it.states; separator=","><\n>}>
<else>
Decision can match input such as "<input>" using multiple alternatives: <conflictingAlts; separator=", ">
<endif>
<if(disabled)><\n>As a result, alternative(s) <disabled; separator=","> were disabled for that input<endif><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by actions.<endif>
>>

DANGLING_STATE(danglingAlts,input) ::= <<
the decision cannot distinguish between alternative(s) <danglingAlts; separator=","> for input such as "<input>"
>>

UNREACHABLE_ALTS(alts) ::= <<
The following alternatives can never be matched: <alts; separator=","><\n>
>>

INSUFFICIENT_PREDICATES(upon,altToLocations,hasPredicateBlockedByAction)
 ::= <<
Input such as "<upon>" is insufficiently covered with predicates at locations: <altToLocations.keys:{alt|alt <alt>: <altToLocations.(alt):{loc| line <loc.line>:<loc.column> at <loc.text>}; separator=", ">}; separator=", "><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by actions.<endif>
>>

DUPLICATE_SET_ENTRY(arg) ::=
 "duplicate token type <arg> when collapsing subrule into set"

ANALYSIS_ABORTED(enclosingRule) ::= <<
ANTLR could not analyze this decision in rule <enclosingRule>; often this is because of recursive rule references visible from the left edge of alternatives.  ANTLR will re-analyze the decision with a fixed lookahead of k=1.  Consider using "options {k=1;}" for that decision and possibly adding a syntactic predicate.
>>

RECURSION_OVERLOW(alt,input,targetRules,callSiteStates) ::= <<
Alternative <alt>: after matching input such as <input> decision
```

```
 cannot predict what comes next due to recursion overflow <targetRules,callSiteStates:{t,c|to <t> from
<c:{s|<s.enclosingRule.name>};separator=", ">}; separator=" and ">
>>

LEFT_RECURSION(targetRules,alt,callSiteStates) ::= <<
Alternative <alt> discovers infinite left-recursion <targetRules,callSiteStates:{t,c|to <t> from
<c:{s|<s.enclosingRule>};separator=", ">}; separator=" and ">
>>

UNREACHABLE_TOKENS(tokens) ::= <<
The following token definitions can never be matched because prior tokens match the same input: <tokens;
separator=",">
>>

TOKEN_NONDETERMINISM(input,conflictingTokens,paths,disabled,hasPredicateBlockedByAction) ::=
<<
<if(paths)>
Decision can match input such as "<input>" using multiple alternatives:
<paths:{ it | alt <it.alt> via NFA path <it.states; separator=","><\n>}>
<else>
Multiple token rules can match input such as "<input>": <conflictingTokens; separator=", "><\n>
<endif>
<if(disabled)><\n>As a result, token(s) <disabled; separator=","> were disabled for
 that input<endif><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by
actions.<endif>
>>

LEFT_RECURSION_CYCLES(listOfCycles) ::= <<
The following sets of rules are mutually left-recursive <listOfCycles:{c| [<c:{r|<r.name>}; separator=", ">]};
separator=" and ">
>>

NONREGULAR_DECISION(ruleName,alts) ::= <<
[fatal] rule <ruleName> has non-LL(*) decision due to recursive rule invocations reachable from alts <alts;
separator=",">.  Resolve by left-factoring or using syntactic predicates or using backtrack=true option.
>>

/* l10n for message levels */
warning() ::= "warning"
error() ::= "error"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/messages/languages/en.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]

```
@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
import org.antlr.runtime.tree.*;<\n>
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
 <parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected TreeAdaptor adaptor = new CommonTreeAdaptor();

public void setTreeAdaptor(TreeAdaptor adaptor) {
 this.adaptor = adaptor;
 <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
}
public
```

```
 TreeAdaptor getTreeAdaptor() {
 return adaptor;
}
>>

@returnScope.ruleReturnMembers() ::= <<
<ASTLabelType> tree;
@Override
public <ASTLabelType> getTree() { return tree; }
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<if(!ruleDescriptor.isSynPred)>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
 ruleDescriptor.wildcardTreeListLabels]:{it | <ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<if(ruleDescriptor.supportsLabelOptimization)>
<ruleDescriptor.allTokenRefsInRewrites
 :{it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInRewrites
 :{it | RewriteRuleSubtreeStream
 stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");}; separator="\n">
<else>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
 :{it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
 :{it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
<endif>
<endif>
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
```

```
<if(outerAlt)>
<if(!rewriteMode && !ruleDescriptor.isSynPred)>
root_0 = (<ASTLabelType>)adaptor.nil();<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions={})
 ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label, ...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label, ...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope)
 ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.getTree());
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(label, {<label>.getTree()})>
```

```
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.getTree());
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(label, {<label>.getTree()})>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel,
 enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
<endif>
<prevRuleRootRef()>.tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
input.replaceChildren(adaptor.getParent(retval.start),
	adaptor.getChildIndex(retval.start),
	adaptor.getChildIndex(_last),
	retval.tree);
<endif>
<endif>
<!
 if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
 :{it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>);};
 separator="\n"
>
<referencedTokenListLabels
 :{it | RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
 separator="\n"
>
<referencedWildcardLabels
 :{it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
 separator="\n"
>
<referencedWildcardListLabels
 :{it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",list_<it>);};
 separator="\n"
>
<referencedRuleLabels
 :{it | RewriteRuleSubtreeStream
 stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>",<it>!=null?<it>.getTree():null);};
 separator="\n"
>
<referencedRuleListLabels
 :{it | RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it>",list_<it>);};
 separator="\n"
>
>>
```

```
/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext()}; separator="||"> ) {
 <alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el
 | stream_<el>.hasNext()}; separator="||"> ) {
 <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext()}; separator="||">) ) {
 throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
 <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
```

```
<if(a.pred)>
if (<a.pred>) {
 <a.alt>
}<\n>
<else>
{
 <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.addChild(root_<enclosingTreeLevel>,
 root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,args,terminalOptions={}) ::= <<
adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());
>>

/** Gen ^($label ...) */
```

```
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args]
 ...) */
rewriteTokenRefRoot(token,elementIndex,args,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);
>>

rewriteImaginaryTokenRef(args,token,elementIndex,terminalOptions={}) ::= <<
adaptor.addChild(root_<treeLevel>, <createImaginaryNode(token,args,terminalOptions)>);
>>

rewriteImaginaryTokenRefRoot(args,token,elementIndex,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<createImaginaryNode(token,args,terminalOptions)>,
root_<treeLevel>);
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule)
 ::= <<
adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>);
>>

rewriteNodeAction(action) ::= <<
adaptor.addChild(root_<treeLevel>, <action>);
>>
```

```
rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<action>, root_<treeLevel>);
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label)
 ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());
>>



createImaginaryNode(tokenType,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
%>

createRewriteNodeFromElement(token,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
```

<endif>
%>

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/AST.stg
No license file was found, but licenses were detected in source scan.

/** Templates for building ASTs during tree parsing.
*
* Deal with many combinations. Dimensions are:
* Auto build or rewrite
*   no label, label, list label (label/no-label handled together)
*   child, root
*   token, set, rule, wildcard
*
* Each combination has its own template except that label/no label
* is combined into tokenRef, ruleRef, ...
*/

```
/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var _first_0:<ASTLabelType>
 = null;
var _last:<ASTLabelType> = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = <ASTLabelType>(_first_0);
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
   retval.tree = <ASTLabelType>(adaptor.getParent(retval.tree));
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = <ASTLabelType>(input.LT(1));
{
var _save_last_<treeLevel>:<ASTLabelType> = _last;
var _first_<treeLevel>:<ASTLabelType> = null;
<if(!rewriteMode)>
var root_<treeLevel>:<ASTLabelType> = <ASTLabelType>(adaptor.nil());
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
   matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions)
 ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
```

```
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions)
 ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>_tree, root_<treeLevel>));
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupTree(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else>
 <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug)
 ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>_tree, root_<treeLevel>));
<if(backtracking)>}<endif>
<endif>
}, ...
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
```

```
adaptor.addChild(root_<treeLevel>, <label>.tree);
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) auto
 construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_<treeLevel> =
<ASTLabelType>(adaptor.becomeRoot(<label>.tree, root_<treeLevel>));
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefRootTrack(...)>
>>

/**
```

```
 ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = <ASTLabelType>(adaptor.rulePostProcessing(root_0));
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ActionScript/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
* Copyright (c) 2005-2008 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2008-2010 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.

* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
 OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CSharp3Target.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports()>
import stringtemplate3
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnInit() ::= <<
self.st = None
>>

@returnScope.ruleReturnMembers() ::= <<
def getTemplate(self):
   return self.st

def toString(self):
   if self.st is not None:
       return self.st.toString()
   return None
__str__
 = toString

>>

@genericParser.init() ::= <<
<@super.init()>
self.templateLib = stringtemplate3.StringTemplateGroup(
   '<name>Templates', lexer='angle-bracket'
   )

>>

@genericParser.members() ::= <<
<@super.members()>
def setTemplateLib(self, templateLib):
   self.templateLib = templateLib

def getTemplateLib(self):
   return self.templateLib

>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
```

```
<ruleRef(...)>
<listLabel(label, {<label>.st})>
>>

rewriteTemplate(alts) ::= <<
# TEMPLATE REWRITE
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
   <first(alts):rewriteTemplateAltFirst()>
   <rest(alts):{it | el<rewriteTemplateAlt(it)>}>
   <if(rewriteMode)><replaceTextInLine()><endif>

<else>
<first(alts):rewriteTemplateAltFirst()>
<rest(alts):{it | el<rewriteTemplateAlt(it)>}>
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
self.input.getTokenStream().replace(

  self.input.getTreeAdaptor().getTokenStartIndex(retval.start),
   self.input.getTreeAdaptor().getTokenStopIndex(retval.start),
   retval.st
   )
<else>
self.input.replace(
   retval.start.getTokenIndex(),
   self.input.LT(-1).getTokenIndex(),
   retval.st
   )
<endif>
>>

rewriteTemplateAltFirst(alt) ::= <<
<if(alt.pred)>
if <alt.pred>:
   # <alt.description>
   retval.st = <alt.alt>
<\n>
<else>
# <alt.description>
retval.st = <alt.alt>
<\n>
<endif>
>>
```

```
rewriteTemplateAlt(alt) ::= <<
<if(alt.pred)>if <alt.pred>:
  # <alt.description>
  retval.st = <alt.alt>
<\n>
<else>se:
  # <alt.description>
  retval.st = <alt.alt>
<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
None
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args)
 ::= <%
self.templateLib.getInstanceOf("<name>"<if(args)>, attributes={<args:{a | "<a.name>": <a.value>}; separator=",
">}<endif>)
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
self.templateLib.getInstanceOf(<expr><if(args)>, attributes={<args:{a | "<a.name>": <a.value>}; separator=",
">}<endif>)
%>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <%
stringtemplate3.StringTemplate("<template>", group=self.templateLib<if(args)>, attributes={<args:{a |
"<a.name>": <a.value>}; separator=", ">}<endif>)
%>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
```

```
(<st>)["<attrName>"] = <expr>
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
stringtemplate3.StringTemplate(<stringExpr>, group=self.templateLib)
>>
```

Found
 in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ST.stg
No license file was found, but licenses were detected in source scan.

```
csharpVisibilityMap ::= [
 "private":"private",
 "protected":"protected",
 "public":"public",
 "fragment":"private",
 default:"private"
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile( LEXER,PARSER,TREE_PARSER, actionScope, actions,
   docComment,
 recognizer,
   name, tokens, tokenNames, rules, cyclicDFAs,
   bitsets, buildTemplate, buildAST, rewriteMode, profile,
   backtracking, synpreds, memoize, numRules,
   fileName, ANTLRVersion, generatedTimestamp, trace,
   scopes, superClass, literals) ::=
<<
//------------------------------------------------------------------------------
// \<auto-generated>
//     This code was generated by a tool.
//     ANTLR Version: <ANTLRVersion>
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// \</auto-generated>
//------------------------------------------------------------------------------

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<if(trace)>
#define ANTLR_TRACE
<endif>
<@debugPreprocessor()>
// The variable 'variable' is assigned but its value is never used.
#pragma warning disable 168, 219
// Unreachable code detected.
#pragma warning disable 162
// Missing XML comment for publicly visible type or member
 'Type_or_Member'
#pragma warning disable 1591

<actions.(actionScope).header>

<@imports>
using System.Collections.Generic;
using Antlr.Runtime;
using Antlr.Runtime.Misc;
```

```
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
using ConditionalAttribute = System.Diagnostics.ConditionalAttribute;
<@end>
<if(actions.(actionScope).namespace)>
namespace <actions.(actionScope).namespace>
{
<endif>
<docComment>
<recognizer>
<if(actions.(actionScope).namespace)>

} // namespace <actions.(actionScope).namespace>
<endif>
>>

lexerInputStreamType() ::= <<
<actions.(actionScope).inputStreamType; null="ICharStream">
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
superClass={<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Lexer
<endif>}) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar,
 actions=actions)> partial class <grammar.recognizerName> : <@superClassName><superClass><@end>
{
 <tokens:{it|public const int <it.name; format="id">=<it.type>;}; separator="\n">
 <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <actions.lexer.members>

   // delegates
   <grammar.delegates:
        {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
   <last(grammar.delegators):{g|private <g.recognizerName> gParent;}>

 <actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>()<! needed by subclasses !>
 {
  OnCreated();
 }

 <actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType()>
input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}> )
  : this(input, new RecognizerSharedState()<grammar.delegators:{g|,
 <g:delegateName()>}>)
 {
 }

 <actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType()> input,
RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
  : base(input, state)
 {
<if(memoize)>
<if(grammar.grammarIsRoot)>
  state.ruleMemo = new System.Collections.Generic.Dictionary\<int, int>[<numRules>+1];<\n><! index from 1..n !>
<endif>
<endif>
  <grammar.directDelegates:
   {g|<g:delegateName()> = new <g.recognizerName>(input, this.state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
  <grammar.delegators:
   {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>

  OnCreated();
 }
 public override string GrammarFileName { get { return "<fileName>"; } }

 private static readonly bool[] decisionCanBacktrack = new bool[0];

<if(grammar.hasDelegates)>
 public override <lexerInputStreamType()> CharStream
 {
  get
  {
   return
 base.CharStream;
  }
  set
  {
   base.CharStream = value;
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
   <grammar.delegators:
    {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
   <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
  }
 }
```

```
<if(grammar.delegates)>
 public override void SetState(RecognizerSharedState state)
 {
  base.SetState(state);
  <grammar.delegates:{g|<g:delegateName()>.SetState(state);}; separator="\n">
 }
<endif>

<endif>
<if(filterMode)>
 <filteringNextToken()>
<endif>

 [Conditional("ANTLR_TRACE")]
 protected virtual void OnCreated() {}
 [Conditional("ANTLR_TRACE")]
 protected virtual void EnterRule(string ruleName, int ruleIndex) {}
 [Conditional("ANTLR_TRACE")]
 protected virtual void LeaveRule(string ruleName, int ruleIndex) {}

   <rules; separator="\n">

 <insertLexerSynpreds(synpreds)>

 #region
 DFA
 <cyclicDFAs:{dfa | DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

 protected override void InitDFAs()
 {
  base.InitDFAs();
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>(this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif>);}; separator="\n">
 }

 <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
 #endregion

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
```

```
public override IToken NextToken()
{
 while (true)
 {
  if (input.LA(1) == CharStreamConstants.EndOfFile)
  {
   IToken eof = new CommonToken((ICharStream)input, CharStreamConstants.EndOfFile, TokenChannels.Default,
input.Index,
 input.Index);
   eof.Line = Line;
   eof.CharPositionInLine = CharPositionInLine;
   return eof;
  }
  state.token = null;
  state.channel = TokenChannels.Default;
  state.tokenStartCharIndex = input.Index;
  state.tokenStartCharPositionInLine = input.CharPositionInLine;
  state.tokenStartLine = input.Line;
  state.text = null;
  try
  {
   int m = input.Mark();
   state.backtracking=1;<! means we won't throw slow exception !>
   state.failed=false;
   mTokens();
   state.backtracking=0;
   <! mTokens backtracks with synpred at backtracking==2
      and we set the synpredgate to allow actions at level 1. !>
   if (state.failed)
   {
    input.Rewind(m);
    input.Consume();<! advance one char and try again !>
   }
   else
   {
    Emit();
    return state.token;
   }
  }
  catch (RecognitionException re)
  {
   // shouldn't happen in backtracking mode, but...
   ReportError(re);
   Recover(re);
  }
 }
}
```

```
public override void Memoize(IIntStream input, int ruleIndex, int ruleStartIndex)
{
 if
 (state.backtracking > 1)
  base.Memoize(input, ruleIndex, ruleStartIndex);
}

public override bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
 if (state.backtracking > 1)
  return base.AlreadyParsedRule(input, ruleIndex);

 return false;
}
>>

actionGate() ::= "state.backtracking == 0"

filteringActionGate() ::= "state.backtracking == 1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
            labelType, members, rewriteElementType,
            filterMode, ASTLabelType="object") ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar, actions=actions)> partial class <grammar.recognizerName> :
<@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
 internal static readonly string[] tokenNames = new string[] {
  "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames;
 separator=", ">
 };
<endif>
 <tokens:{it|public const int <it.name; format="id">=<it.type>;}; separator="\n">

<if(grammar.delegates)>
 // delegates
 <grammar.delegates:
   {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
<endif>
<if(grammar.delegators)>
 // delegators
 <grammar.delegators:
   {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
 <last(grammar.delegators):{g|private <g.recognizerName> gParent;}>
```

```
<endif>

<if(grammar.delegates)>
 public override void SetState(RecognizerSharedState state)
 {
  base.SetState(state);
  <grammar.delegates:{g|<g:delegateName()>.SetState(state);}; separator="\n">
 }

<if(TREE_PARSER)>
 public override void SetTreeNodeStream(ITreeNodeStream input)
 {
  base.SetTreeNodeStream(input);
  <grammar.delegates:{g|<g:delegateName()>.SetTreeNodeStream(input);}; separator="\n">
 }
<endif>
<endif>

 <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <@members()>

 public override
 string[] TokenNames { get { return <grammar.composite.rootGrammar.recognizerName>.tokenNames; } }
 public override string GrammarFileName { get { return "<fileName>"; } }

 <members>

 [Conditional("ANTLR_TRACE")]
 protected virtual void OnCreated() {}
 [Conditional("ANTLR_TRACE")]
 protected virtual void EnterRule(string ruleName, int ruleIndex) {}
 [Conditional("ANTLR_TRACE")]
 protected virtual void LeaveRule(string ruleName, int ruleIndex) {}

 #region Rules
 <rules; separator="\n">
 #endregion Rules

<if(grammar.delegatedRules)>
<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
 #region Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
 <ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <returnType(ruleDescriptor)>
<ruleDescriptor.name; format="id">(<ruleDescriptor.parameterScope:parameterScope()>) <!throws
RecognitionException !>{ <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name;
```

```
 format="id">(<if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope.attributes:{a|<a.name;
format="id">}; separator=", "><endif>); \}}; separator="\n">
 #endregion Delegated rules
<endif>

 <insertSynpreds(synpreds)>

<if(cyclicDFAs)>
 #region DFA
 <cyclicDFAs:{dfa | private DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

 protected override void InitDFAs()
 {
  base.InitDFAs();
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>( this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif> );}; separator="\n">
 }

 <cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>
 #endregion DFA
<endif>

<if(bitsets)>
 #region Follow sets
 private static class Follow
 {
  <bitsets:{it|<bitset(name={_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>}; separator="\n">
 }
 #endregion Follow sets
<endif>
}
>>

@genericParser.members() ::= <<
#if ANTLR_DEBUG
private static readonly
 bool[] decisionCanBacktrack =
 new bool[]
 {
  false, // invalid decision
  <grammar.decisions:{d | <d.dfa.hasSynPred>}; wrap="\n", separator=", ">
 };
#else
private static readonly bool[] decisionCanBacktrack = new bool[0];
#endif
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<inputStreamType>
input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
```

```
 : this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>)
{
}
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<inputStreamType> input,
RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
 : base(input, state)
{
<if(grammar.directDelegates)>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
<endif>
<if(grammar.indirectDelegates)>
 <grammar.indirectDelegates:{g
 | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;}; separator="\n">
<endif>
<if(grammar.delegators)>
 <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
<endif>
 <parserCtorBody()>
 OnCreated();
}
>>

// imported grammars are 'public' (can't be internal because their return scope classes must be accessible)
parserModifier(grammar, actions) ::= <<
<if(grammar.grammarIsRoot)><actions.(actionScope).modifier; null="public"><else>public<endif>
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new System.Collections.Generic.Dictionary\<int,
int>[<length(grammar.allImportedRules)>+1];<\n><! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
	ASTLabelType="object",
superClass={<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Parser
<endif>},
 labelType="IToken",
	members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="IToken", ...)>
>>
```

```
/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
         numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="object",
superClass={<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Tree.<
if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif><endif>},
         members={<actions.treeparser.members>}) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*
 nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
[Conditional("ANTLR_TRACE")]
protected virtual void EnterRule_<ruleName>_fragment() {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule_<ruleName>_fragment() {}

// $ANTLR start <ruleName>
<ruleModifier(grammar,ruleDescriptor)> void
<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>)
{
 <ruleLabelDefs()>
 EnterRule_<ruleName>_fragment();
 EnterRule("<ruleName>_fragment", <ruleDescriptor.index>);
 TraceIn("<ruleName>_fragment", <ruleDescriptor.index>);
 try
 {
  <block>
 }
 finally
 {
  TraceOut("<ruleName>_fragment", <ruleDescriptor.index>);
  LeaveRule("<ruleName>_fragment", <ruleDescriptor.index>);
  LeaveRule_<ruleName>_fragment();
 }
}
//
```

```
 $ANTLR end <ruleName>
>>

insertLexerSynpreds(synpreds) ::= <<
<insertSynpreds(synpreds)>
>>

insertSynpreds(synpreds) ::= <<
<if(synpreds)>
#region Synpreds
private bool EvaluatePredicate(System.Action fragment)
{
 bool success = false;
 state.backtracking++;
 <@start()>
 try { DebugBeginBacktrack(state.backtracking);
 int start = input.Mark();
 try
 {
  fragment();
 }
 catch ( RecognitionException re )
 {
  System.Console.Error.WriteLine("impossible: "+re);
 }
 success = !state.failed;
 input.Rewind(start);
 } finally { DebugEndBacktrack(state.backtracking, success); }
 <@stop()>
 state.backtracking--;
 state.failed=false;
 return success;
}
#endregion Synpreds
<endif>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if (state.backtracking > 0 && AlreadyParsedRule(input, <ruleDescriptor.index>)) { <returnFromRule()> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) <returnFromRule()><endif>
>>
```

```
/**
 This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true; <returnFromRule()>}<endif>
>>

ruleWrapperMap ::= [
 "bottomup":{<ruleWrapperBottomup()>},
 "topdown":{<ruleWrapperTopdown()>},
 default:""
]

ruleWrapperBottomup() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Bottomup() { return bottomup(); }
<endif>
>>

ruleWrapperTopdown() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Topdown() { return topdown(); }
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(ruleDescriptor.returnScope)>

[Conditional("ANTLR_TRACE")]
protected
 virtual void EnterRule_<ruleName>() {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule_<ruleName>() {}
<ruleWrapperMap.(ruleName)>
// $ANTLR start "<ruleName>"
// <fileName>:<description>
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <returnType(ruleDescriptor)> <ruleName;
format="id">(<ruleDescriptor.parameterScope:parameterScope()>)
{
 EnterRule_<ruleName>();
 EnterRule("<ruleName>", <ruleDescriptor.index>);
 TraceIn("<ruleName>", <ruleDescriptor.index>);
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
 try { DebugEnterRule(GrammarFileName, "<ruleName>");
 DebugLocation(<ruleDescriptor.tree.line>, <ruleDescriptor.EORNode.charPositionInLine>);
 <@preamble()>
 try
 {
  <ruleMemoization(name=ruleName)>
  <block>
  <ruleCleanUp()>
  <(ruleDescriptor.actions.after):execAction()>
 }
<if(exceptions)>
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
 <actions.(actionScope).rulecatch>
<else>
 catch
 (RecognitionException re)
 {
  ReportError(re);
  Recover(input,re);
 <@setErrorReturnValue()>
 }
<endif>
<endif>
<endif>
 finally
 {
  TraceOut("<ruleName>", <ruleDescriptor.index>);
  LeaveRule("<ruleName>", <ruleDescriptor.index>);
  LeaveRule_<ruleName>();
      <memoize()>
      <ruleScopeCleanUp()>
      <finally>
   }
 DebugLocation(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);
 } finally { DebugExitRule(GrammarFileName, "<ruleName>"); }
 <@postamble()>
 <returnFromRule()><\n>
}
// $ANTLR end "<ruleName>"
>>

// imported grammars need to have internal rules
ruleModifier(grammar,ruleDescriptor) ::= <<
```

```
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>internal<endif>
>>

// imported grammars need to have public return scopes
returnScopeModifier(grammar,ruleDescriptor) ::= <<
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
 null="private"><else>public<endif>
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
 <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType(ruleDescriptor)> retval = new <returnType(ruleDescriptor)>();
retval.Start = (<labelType>)input.LT(1);
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name; format="id"> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.Index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it|<it>_stack.Push(new <it>_scope());<it>_scopeInit(<it>_stack.Peek());};
separator="\n">
<ruleDescriptor.ruleScope:{it|<it.name>_stack.Push(new
<it.name>_scope());<it.name>_scopeInit(<it.name>_stack.Peek());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it|<it>_scopeAfter(<it>_stack.Peek());<it>_stack.Pop();};
 separator="\n">
<ruleDescriptor.ruleScope:{it|<it.name>_scopeAfter(<it.name>_stack.Peek());<it.name>_stack.Pop();};
separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wil
dcardTreeListLabels]
```

```
   :{it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<ruleDescriptor.tokenListLabels
   :{it|List\<<labelType>\> list_<it.label.text> = null;}; separator="\n"
>
<[ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it|List\<<ASTLabelType>\> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:ruleLabelDef(); separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<[ruleDescriptor.charListLabels,
  ruleDescriptor.charLabels]
 :{it|int <it.label.text> = 0;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{it|List\<<labelType>\> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.charListLabels:{it|List\<int> list_<it.label.text> = null;}; separator="\n"
>
>>

returnFromRule() ::= <%
return
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<! This comment is a hack to make sure the following
  single space appears in the output. !> <ruleDescriptor.singleValueReturnName>
<else>
<!!> retval
<endif>
<endif>
<endif>
;
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
```

```
retval.Stop = (<labelType>)input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (state.backtracking > 0) { Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the
 lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<

[Conditional("ANTLR_TRACE")]
protected virtual void EnterRule_<ruleName>() {}
[Conditional("ANTLR_TRACE")]
protected virtual void LeaveRule_<ruleName>() {}

// $ANTLR start "<ruleName>"
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> void
m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>)
{
 EnterRule_<ruleName>();
 EnterRule("<ruleName>", <ruleDescriptor.index>);
 TraceIn("<ruleName>", <ruleDescriptor.index>);
   <ruleScopeSetUp()>
   <ruleDeclarations()>
  try
  {
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
<else>
  int _type = <ruleName>;
  int _channel = DefaultTokenChannel;
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
  <ruleCleanUp()>
  state.type = _type;
  state.channel =
 _channel;
  <(ruleDescriptor.actions.after):execAction()>
<endif>
 }
 finally
 {
  TraceOut("<ruleName>", <ruleDescriptor.index>);
  LeaveRule("<ruleName>", <ruleDescriptor.index>);
  LeaveRule_<ruleName>();
      <ruleScopeCleanUp()>
      <memoize()>
   }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<

public override void mTokens()
{
 <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterSubRule(<decisionNumber>);
try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
<@prebranch()>
switch
 (alt<decisionNumber>)
{
<alts:{a|<altSwitchCase(i,a)>}>
```

```
}
} finally { DebugExitSubRule(<decisionNumber>); }
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
switch (alt<decisionNumber>)
{
<alts:{a|<altSwitchCase(i,a)>}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/**
 A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
```

```
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
 <decision>
 } finally { DebugExitDecision(<decisionNumber>); }
 <@postdecision()>
 switch (alt<decisionNumber>)
 {
 <alts:{a|<altSwitchCase(i,a)>}>
 default:
  if (cnt<decisionNumber> >= 1)
   goto loop<decisionNumber>;

  <ruleBacktrackFailure()>
  EarlyExitException eee<decisionNumber> = new EarlyExitException( <decisionNumber>, input );
  DebugRecognitionException(eee<decisionNumber>);
  <@earlyExitException()>
  throw eee<decisionNumber>;
 }
 cnt<decisionNumber>++;
}
loop<decisionNumber>:
 ;

}
 finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 try { DebugEnterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);
 <decision>
 } finally { DebugExitDecision(<decisionNumber>); }
 <@postdecision()>
 switch ( alt<decisionNumber> )
 {
 <alts:{a|<altSwitchCase(i,a)>}>
 default:
  goto loop<decisionNumber>;
 }
}

loop<decisionNumber>:
 ;

} finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
 */
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
 <@prealt()>
 DebugEnterAlt(<altNum>);
 <alt>
 break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>
```

```
/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(it) ::= <%
<@prematch()>
DebugLocation(<it.line>, <it.pos>);<\n>
<it.el><\n>
%>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={})
 ::= <<
<if(label)><label>=(<labelType>)<endif>Match(input,<token>,Follow._<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

listLabelElem(label,elem,elemType) ::= <<
if (list_<label>==null) list_<label>=new List\<<elemType; null={<labelType>}>\>();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>
```

```
/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<matchSetLabel()>
<endif>
if (<s>)
{
 input.Consume();
 <postmatchCode>
 <if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
}
else
{
 <ruleBacktrackFailure()>
 MismatchedSetException
 mse = new MismatchedSetException(null,input);
 DebugRecognitionException(mse);
 <@mismatchedSetException()>
<if(LEXER)>
 Recover(mse);
 throw mse;
<else>
 throw mse;
 <! use following code to make it recover inline; remove throw mse;
 recoverFromMismatchedSet(input,mse,Follow._set_in_<ruleName><elementIndex>);
 !>
<endif>
}<\n>
>>

matchSetUnchecked(s,label,elementIndex,postmatchCode=false) ::= <%
<if(label)>
<matchSetLabel()><\n>
<endif>
input.Consume();<\n>
<if(postmatchCode)>
<postmatchCode><\n>
<endif>
<if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
%>

matchSetLabel() ::= <%
<if(LEXER)>
<label>= input.LA(1);
<else>
<label>=(<labelType>)input.LT(1);
<endif>
%>
```

```
matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/**
 Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <%
<if(label)>
int <label>Start = CharIndex;<\n>
Match(<string>); <checkRuleBacktrackFailure()><\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<label> = new <labelType>(input, TokenTypes.Invalid, TokenChannels.Default, <label>Start, CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
%>

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<wildcard(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex)
 ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
```

```
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(Follow._<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name; format="id">(<args;
separator=", ">);
PopFollow();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem=label,elemType={<ASTLabelType>},...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was
 the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()><\n>
<label> = new <labelType>(input, TokenTypes.Invalid, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
%>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
```

```
<lexerRuleRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/**
 EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
Match(EOF); <checkRuleBacktrackFailure()><\n>
<labelType> <label> = new <labelType>(input, EOF, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
%>

// used for left-recursive rules
recRuleDefArg()                       ::= "int <recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root,
 actionsAfterRoot, children, nullableChildList,
   enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down)
{
 Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
>>
```

```
/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if (!(<evalPredicate(...)>))
{
 <ruleBacktrackFailure()>
 throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges;
 separator="\nelse ">
<if((!isTrue.(last(edges).labelExpr)) && (!last(edges).predicates))>
else
{
<if(eotPredictsAlt)>
 alt<decisionNumber> = <eotPredictsAlt>;
<else>
 <ruleBacktrackFailure()>
 NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
 DebugRecognitionException(nvae);
 <@noViableAltException()>
 throw nvae;
<endif>
}
<endif>
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
 *  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt>;<! if no edges, don't gen ELSE !>
<else>
else
{
 alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> && (<predicates>)<endif>)
{
 <targetState>
}
>>

//
 F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
 alt<decisionNumber>=<eotPredictsAlt>;
 break;<\n>
<else>
 {
  <ruleBacktrackFailure()>
  NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
  DebugRecognitionException(nvae);
  <@noViableAltException()>
  throw nvae;
 }
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
<if(eotPredictsAlt)>
default:
 alt<decisionNumber>=<eotPredictsAlt>;
 break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels,
 targetState) ::= <<
<labels:{it|case <it>:}; separator="\n">
 {
 <targetState>
 }
 break;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
try
{
 alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
```

```
}
catch (NoViableAltException nvae)
{
 DebugRecognitionException(nvae);
 throw;
}
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa)
 ::= <<
private class DFA<dfa.decisionNumber> : DFA
{
 private const string DFA<dfa.decisionNumber>_eotS =
  "<dfa.javaCompressedEOT; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_eofS =
  "<dfa.javaCompressedEOF; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_minS =
  "<dfa.javaCompressedMin; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_maxS =
  "<dfa.javaCompressedMax; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_acceptS =
  "<dfa.javaCompressedAccept; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_specialS =
  "<dfa.javaCompressedSpecial; wrap="\"+\n\t\t\"">}>";
 private static readonly string[] DFA<dfa.decisionNumber>_transitionS =
  {
   <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
  };

 private static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
 private static readonly short[]
 DFA<dfa.decisionNumber>_eof = DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
 private static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
 private static readonly char[] DFA<dfa.decisionNumber>_max =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
 private static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
 private static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
 private static readonly short[][] DFA<dfa.decisionNumber>_transition;

 static DFA<dfa.decisionNumber>()
 {
  int numStates = DFA<dfa.decisionNumber>_transitionS.Length;
  DFA<dfa.decisionNumber>_transition = new short[numStates][];
  for ( int i=0; i \< numStates; i++ )
  {
   DFA<dfa.decisionNumber>_transition[i] =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
  }
 }

 public
 DFA<dfa.decisionNumber>( BaseRecognizer recognizer<if(dfa.specialStateSTs)>, SpecialStateTransitionHandler
specialStateTransition<endif> )
<if(dfa.specialStateSTs)>
  : base(specialStateTransition)
<endif>
 {
  this.recognizer = recognizer;
  this.decisionNumber = <dfa.decisionNumber>;
  this.eot = DFA<dfa.decisionNumber>_eot;
  this.eof = DFA<dfa.decisionNumber>_eof;
  this.min = DFA<dfa.decisionNumber>_min;
  this.max = DFA<dfa.decisionNumber>_max;
  this.accept = DFA<dfa.decisionNumber>_accept;
  this.special = DFA<dfa.decisionNumber>_special;
  this.transition = DFA<dfa.decisionNumber>_transition;
 }

 public override string Description { get { return "<dfa.description>"; } }

 public override void Error(NoViableAltException nvae)
 {
  DebugRecognitionException(nvae);
 }
}<\n>
<if(dfa.specialStateSTs)>
private int SpecialStateTransition<dfa.decisionNumber>(DFA dfa, int s, IIntStream _input)<! throws
NoViableAltException!>
{
 <if(LEXER)>
 IIntStream input = _input;
 <endif>
 <if(PARSER)>
 ITokenStream
 input = (ITokenStream)_input;
 <endif>
 <if(TREE_PARSER)>
 ITreeNodeStream input = (ITreeNodeStream)_input;
 <endif>
 int _s = s;
 s = -1;
 <! pull these outside the switch cases to save space on locals !>
 int LA<dfa.decisionNumber>_1 = input.LA(1);
 int index<dfa.decisionNumber>_1 = input.Index;
 switch (_s)
 {
 <dfa.specialStateSTs:{state |case <i0>:<! compressed special state numbers 0..n-1 !>
 <state>}; separator="\n">

 default:
  break;
 }

 if (s >= 0)
  return s;

<if(backtracking)>
 if (state.backtracking > 0) {state.failed=true; return -1;}
<endif>
 NoViableAltException nvae = new NoViableAltException(dfa.Description, <dfa.decisionNumber>, _s, input);
 dfa.Error(nvae);
 throw nvae;
}
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
<if(semPredState)>
 <! get next lookahead symbol to test edges, then rewind !>
 input.Rewind();
<endif>
 <edges;
 separator="\nelse ">
<if(semPredState)>
 <! return input cursor to state before we rewound !>
 input.Seek(index<decisionNumber>_1);
<endif>
 break;
```

```
}
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> && (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "EvaluatePredicate(<pred>_fragment)"

lookaheadTest(atom,k,atomAsInt)
 ::= "LA<decisionNumber>_<k>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<k><ge()><lower> && LA<decisionNumber>_<k><le()><upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::=
"(input.LA(<k>)<ge()><lower> && input.LA(<k>)<le()><upper>)"

le() ::= "\<="
ge() ::= ">="
```

```
setTest(ranges) ::= <<
<ranges; separator="||">
>>

// A T T R I B U T E S

attributeScope(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
protected sealed partial class <scope.name>_scope
{
 <scope.attributes:{it|public <it.decl>;}; separator="\n">
}
<if(scope.actions.scopeinit)>
protected void <scope.name>_scopeInit( <scope.name>_scope scope )
{
 <scope.actions.scopeinit>
}
<else>
protected virtual void <scope.name>_scopeInit(
 <scope.name>_scope scope ) {}
<endif>
<if(scope.actions.scopeafter)>
protected void <scope.name>_scopeAfter( <scope.name>_scope scope )
{
 <scope.actions.scopeafter>
}
<else>
protected virtual void <scope.name>_scopeAfter( <scope.name>_scope scope ) {}
<endif>
protected readonly ListStack\<<scope.name>_scope> <scope.name>_stack = new
ListStack\<<scope.name>_scope>();
<endif>
<endif>
>>

globalAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

ruleAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

returnStructName(it) ::= "<it.name>_return"

returnType(ruleDescriptor) ::= <%
<if(ruleDescriptor.returnScope.attributes && ruleDescriptor.hasMultipleReturnValues)>
```

```
 <ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<elseif(ruleDescriptor.hasMultipleReturnValues)>
 <ruleReturnBaseType()>
<elseif(ruleDescriptor.hasSingleReturnValue)>
 <ruleDescriptor.singleValueReturnType>
<else>
 void
<endif>
%>

/** Generate the C# type associated with
 a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
 <ruleReturnBaseType()>
<elseif(referencedRule.hasSingleReturnValue)>
 <referencedRule.singleValueReturnType>
<else>
 void
<endif>
%>

delegateName(it) ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
default(<typeName>)
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;
%>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(scope.attributes && ruleDescriptor.hasMultipleReturnValues)>
<returnScopeModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)>
```

```
 sealed partial class <ruleDescriptor:returnStructName()> : <ruleReturnBaseType()><@ruleReturnInterfaces()>
{
 <scope.attributes:{it|public <it.decl>;}; separator="\n">
 <@ruleReturnMembers()>
}
<endif>
>>

ruleReturnBaseType() ::= <%
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<labelType>>
%>

@returnScope.ruleReturnMembers() ::= <<
>>

parameterScope(scope) ::= <<
<scope.attributes:{it|<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= <<
<attr.name; format="id">
>>

parameterSetAttributeRef(attr,expr) ::= <<
<attr.name; format="id"> =<expr>;
>>

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id">
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id">
<else>
<scope>_stack.Peek().<attr.name; format="id">
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::=
 <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id"> = <expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id"> = <expr>;
<else>
<scope>_stack.Peek().<attr.name; format="id"> = <expr>;
```

```
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.Count>0 && $function::name.Equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?((<returnType(referencedRule)>)<scope>).<attr.name; format="id">:<initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id">
<else>
<attr.name; format="id">
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr)
 ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id"> =<expr>;
<else>
<attr.name; format="id"> =<expr>;
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.Text:default(string))"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.Type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.Line:0)"
```

```
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.CharPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.Channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.TokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int.Parse(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr)
 ::= "(<scope>!=null?((<labelType>)<scope>.Start):default(<labelType>))"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.Stop):default(<labelType>))"
ruleLabelPropertyRef_tree(scope,attr) ::=
"(<scope>!=null?((<ASTLabelType>)<scope>.Tree):default(<ASTLabelType>))"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
(<scope>!=null?(input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(<scope>.Start),
 input.TreeAdaptor.GetTokenStopIndex(<scope>.Start))):default(string))
<else>
(<scope>!=null?input.ToString(<scope>.Start,<scope>.Stop):default(string))
<endif>
%>

ruleLabelPropertyRef_st(scope,attr) ::= "(<scope>!=null?<scope>.Template:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(<scope>!=null?<scope>.Type:0)"

lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(<scope>!=null?<scope>.Line:0)"

lexerRuleLabelPropertyRef_pos(scope,attr)
 ::=
   "(<scope>!=null?<scope>.CharPositionInLine:-1)"

lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(<scope>!=null?<scope>.Channel:0)"

lexerRuleLabelPropertyRef_index(scope,attr) ::=
   "(<scope>!=null?<scope>.TokenIndex:0)"

lexerRuleLabelPropertyRef_text(scope,attr) ::=
   "(<scope>!=null?<scope>.Text:default(string))"

lexerRuleLabelPropertyRef_int(scope,attr) ::=
   "(<scope>!=null?int.Parse(<scope>.Text):0)"
```

```
// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.Start"
rulePropertyRef_stop(scope,attr) ::= "retval.Stop"
rulePropertyRef_tree(scope,attr) ::= "retval.Tree"
rulePropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start))
<else>
input.ToString(retval.Start,input.LT(-1))
<endif>
%>
rulePropertyRef_st(scope,attr)
 ::= "retval.Template"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.Tree = <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.Template =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <%
<if(backtracking)>
if (<actions.(actionScope).synpredgate>)<\n>
{<\n>
<@indentedAction()><\n>
}
<else>
<action>
<endif>
%>

@execAction.indentedAction()
 ::= <<
 <action>
>>
```

```
/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static readonly BitSet <name> = new BitSet(new ulong[]{<words64:{it|<it>UL};separator=",">});
>>

codeFileExtension() ::= ".cs"

true_value() ::= "true"
false_value() ::= "false"

isTrue ::= [
 "true" : true,
 default : false
]
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/CSharp2.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

```
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<if(!ruleDescriptor.isSynPred)>
<ASTLabelType>
 _first_0 = null;
<ASTLabelType> _last = null;<\n>
<endif>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel=false, treeLevel=false) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = _first_0;
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
 retval.tree = (<ASTLabelType>)adaptor.getParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
  enclosingTreeLevel, treeLevel) ::= <<
<if(!ruleDescriptor.isSynPred)>
```

```
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel>
 = (<ASTLabelType>)adaptor.nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> =
(<ASTLabelType>)<root.el.label>.getTree();
<elseif(root.el.label)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
 match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
 <children:element()>
 match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
<else>
<super.tree(...)>
<endif>
>>

// TOKEN AST STUFF

/** ID!
 and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
```

```
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<else>
<super.tokenRefBang(...)>
<endif>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
<else>
<super.tokenRef(...)>
<endif>
>>

/** label+=TOKEN auto
 construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRef(...)>
<listLabel(elem=label,...)>
<else>
<super.tokenRefAndListLabel(...)>
<endif>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
<else>
<super.tokenRefRuleRoot(...)>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
<else>
<super.tokenRefRuleRootAndListLabel(...)>
<endif>
>>

/**
 Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupTree(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
<else>
<super.wildcard(...)>
<endif>
>>

// SET AST

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
```

```
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree
 = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSet(...)>
<endif>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
<else>
<super.matchRuleBlockSet(...)>
<endif>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
<else>
<super.matchSetBang(...)>
<endif>
>>

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(!ruleDescriptor.isSynPred)>
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree
 = new <terminalOptions.node>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSetRuleRoot(...)>
<endif>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.getTree());
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = (<ASTLabelType>)<label>.getTree();
<endif>
<else>
<super.ruleRef(...)>
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRef(...)>
<listLabel(label, {<label>.getTree()})>
<else>
<super.ruleRefAndListLabel(...)>
<endif>
>>

/**
 ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(), root_<treeLevel>);
<endif>
```

```
<else>
<super.ruleRefRuleRoot(...)>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.getTree()})>
<else>
<super.ruleRefRuleRootAndListLabel(...)>
<endif>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::=
 <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
<else>
<super.ruleRefTrackAndListLabel(...)>
<endif>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
<else>
<super.ruleRefRootTrack(...)>
<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
<else>
```

```
<super.ruleRefRuleRootTrackAndListLabel(...)>
<endif>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,args,terminalOptions={}) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::=
 <<
<super.ruleCleanUp()>
<if(!ruleDescriptor.isSynPred)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/ASTTreeParser.stg

No license file was found, but licenses were detected in source scan.

/*

[The "BSD license"]

Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC

http://www.temporal-wave.com

http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/

@rule.setErrorReturnValue()
 ::= <<
retval.tree = (<ASTLabelType>)(ADAPTOR->errorNode(ADAPTOR, INPUT, retval.start, LT(-1), EXCEPTION));
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
ADAPTOR->addChild(ADAPTOR, root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
```

```
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_0));
<if(backtracking)>}<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions)
 ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>



/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>



/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,terminalOptions,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus
 ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )
<endif>ADAPTOR->addChild(ADAPTOR, root_0, <createNodeFromToken(...)>);},...)>
>>
```

```
matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,terminalOptions,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(label)>
<label>=(<labelType>)LT(1);<\n>
<endif>
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0 =
(<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <createNodeFromToken(...)>,
 root_0));},...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>ADAPTOR->addChild(ADAPTOR, root_0,
<label>.tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0 = (<ASTLabelType>)(ADAPTOR-
>becomeRoot(ADAPTOR, <label>.tree, root_0));
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelAST(...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabelAST(...)>
>>
```

```
/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRefRuleRoot(...)>
<listLabelAST(...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
ADAPTOR->addChild(ADAPTOR, root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

wildcardBang(token,label,elementIndex,terminalOptions) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
root_0 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_0));
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,terminalOptions) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>New(<label>) <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>))
<endif>
>>

ruleCleanUp()
 ::= <<
<super.ruleCleanUp()>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{<\n>
<endif>
<if(!ruleDescriptor.isSynPred)>
 retval.stop = LT(-1);<\n>
<endif>
 retval.tree = (<ASTLabelType>)(ADAPTOR->rulePostProcessing(ADAPTOR, root_0));
 ADAPTOR->setTokenBoundaries(ADAPTOR, retval.tree, retval.start, retval.stop);
```

```
   <ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
   <ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<if(backtracking)>
}<\n>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/ASTParser.stg

No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2006, 2007 Kay Roepke 2010 Alan Condit
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
/*
*  Template group file for the Objective C code generator.
*  Heavily based on Java.stg
*
*  Written by Kay Roepke <kroepke(at)classdump.org>
*  Modified by Alan Condit <acondit(at)ipns.com>
*
*  This file is part of ANTLR and subject to the same license as ANTLR itself.
*/

objcTypeInitMap ::= [
   "int"          : "0",             // Integers     start out being 0
   "long"         : "0",             // Longs        start out being 0
   "float"
         : "0.0",            // Floats       start out being 0
   "double"       : "0.0",           // Doubles      start out being 0
   "BOOL"         : "NO",            // Booleans     start out being Antlr ObjC for false
   "byte"         : "0",             // Bytes        start out being 0
   "short"        : "0",             // Shorts       start out being 0
   "char"         : "0",             // Chars        start out being 0
   "id"           : "nil",           // ids          start out being nil
   default        : "nil"            // anything other than an atomic type
]

// System.Boolean.ToString() returns "True" and "False", but the proper C# literals are "true" and "false"
// The Java version of Boolean returns "true" and "false", so they map to themselves here.
booleanLiteral ::= [
 "True":"true",
 "False":"false",
 "true":"YES",
 "false":"NO",
 default:"NO"
]



className() ::= "<name><!<if(LEXER)>Lexer<else><if(TREE_PARSER)>Tree<endif>Parser<endif>!>"
leadIn(type)
 ::=
<<
/** \file
*  This <type> file was generated by $ANTLR version <ANTLRVersion>
*
*     -  From the grammar source file : <fileName>
*     -                            On : <generatedTimestamp>
<if(LEXER)>
*     -                 for the lexer : <name>Lexer
<endif>
<if(PARSER)>
*     -                for the parser : <name>Parser
<endif>
<if(TREE_PARSER)>
*     -           for the tree parser : <name>TreeParser
<endif>
```

```
*
* Editing it, at least manually, is not wise.
*
* ObjC language generator and runtime by Alan Condit, acondit|hereisanat|ipns|dotgoeshere|com.
*
*
>>

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile( LEXER,
          PARSER,
          TREE_PARSER,
          actionScope,
          actions,
          docComment,
          recognizer,
          name,
          tokens,
          tokenNames,
          rules,
          cyclicDFAs,
          bitsets,
           buildTemplate,
          buildAST,
          rewriteMode,
          profile,
          backtracking,
          synpreds,
          memoize,
          numRules,
          fileName,
          ANTLRVersion,
          generatedTimestamp,
          trace,
          scopes,
          superClass,
          literals
          ) ::=
<<
<leadIn("OBJC source")>
*/
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<! <if(actions.(actionScope).header)>
/* ==========================================================================
* This is what the grammar programmer asked us to put at the top of every file.
```

```
*/
<actions.(actionScope).header>
/* End of Header action.
* ==============================================================================
*/
<endif> !>

/* -----------------------------------------
* Include the ANTLR3 generated header file.
*/
#import "<name><!<if(LEXER)>Lexer<else><if(TREE_PARSER)>Tree<endif>Parser<endif>!>.h"
<actions.(actionScope).postinclude>
/*
 ----------------------------------------- */

<docComment>

<if(literals)>
/** String literals used by <name> that we must do things like MATCHS() with.
*  C will normally just lay down 8 bit characters, and you can use L"xxx" to
*  get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
*  we perform this little trick of defining the literals as arrays of UINT32
*  and passing in the address of these.
*/
<literals:{it | static ANTLR3_UCHAR  lit_<i>[]  = <it>;}; separator="\n">

<endif>

/* ============================================================================= */
/* =============================================================================
* Start of recognizer
*/
<recognizer>
>>
headerFileExtension() ::= ".h"

headerFile( LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,

       tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,
            rewriteMode,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,
            generatedTimestamp,
            trace,
            scopes,
            superClass,
            literals
          ) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
<actions.(actionScope).preincludes>
/* =============================================================================
* Standard antlr OBJC runtime definitions
*/
#import \<Cocoa/Cocoa.h>
#import \<ANTLR/ANTLR.h>
/* End of standard antlr3 runtime definitions
* =============================================================================
*/
<actions.(actionScope).includes>
<@end>

<if(LEXER)>
<lexerHeaderFile(...)>
<endif>
<if(PARSER)>
<parserHeaderFile(...)>
<endif>
<if(TREE_PARSER)>
<treeParserHeaderFile(...)>
<endif>
<docComment>
>>

lexerHeaderFile(
    LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,
            generatedTimestamp,
            trace,
            scopes,
            superClass="Lexer"
            ) ::=
<<

<if(actions.(actionScope).header)>
/* =============================================================================
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =============================================================================
    */
<endif>

/* Start cyclicDFAInterface */
<cyclicDFAs:cyclicDFAInterface()>

#pragma mark Rule return scopes Interface start
<rules:{rule |
<rule.ruleDescriptor:{ruleDescriptor | <returnScopeInterface(scope=ruleDescriptor.returnScope)>}>}>
#pragma mark Rule return scopes Interface end
#pragma mark Tokens
#ifndef TOKENLISTAlreadyDefined
#define TOKENLISTAlreadyDefined 1
```

```
#ifdef EOF
#undef EOF
#endif
<tokens:{it | #define <it.name> <it.type>}; separator="\n">
#endif
/* interface lexer class */
@interface <className()> <@superClassName>: <superClass><@end> { // line 283
<cyclicDFAs:{dfa | DFA<dfa.decisionNumber> *dfa<dfa.decisionNumber>;}; separator="\n">
<synpreds:{pred | SEL <pred>Selector;}; separator="\n">
/* ObjC start of actions.lexer.memVars */
<actions.lexer.memVars>
/* ObjC end of actions.lexer.memVars */
}
+ (void) initialize;
+ (<className()> *)new<className()>WithCharStream:(id\<CharStream>)anInput;
/* ObjC start actions.lexer.methodsDecl */
<actions.lexer.methodsDecl>
/* ObjC
 end actions.lexer.methodsDecl */
<rules:{rule |
- (<rule.ruleDescriptor:{ruleDescriptor|<returnType()>}>)
<if(!rule.ruleDescriptor.isSynPred)>m<rule.ruleName><else><rule.ruleName>_fragment<endif>
<if(rule.ruleDescriptor.parameterScope)><rule.ruleDescriptor.parameterScope:parameterScope()><endif>; };
separator="\n"><\n>
@end /* end of <className()> interface */<\n>
>>

headerReturnScope(ruleDescriptor) ::= "<returnScopeInterface(...)>"
headerReturnType(ruleDescriptor) ::= <<
<if(LEXER)>
<if(!r.ruleDescriptor.isSynPred)>
void
<else>
<ruleDescriptor:returnType()>
<endif>
<else>
<ruleDescriptor:returnType()>
<endif>
>>
// Produce the lexer output
lexer(  grammar,
      name,
      tokens,
      scopes,
      rules,
      numRules,
      filterMode,
      labelType="CommonToken",
    superClass="Lexer"
    ) ::= <<
<cyclicDFAs:cyclicDFA()>

/** As per Terence: No returns for lexer rules! */
<!
#pragma mark Rule return scopes start
<rules:{rule | <rule.ruleDescriptor:{ruleDescriptor
 |
<returnScopeImplementation(scope=ruleDescriptor.returnScope)>}>
}>
#pragma mark Rule return scopes end
!>
@implementation <grammar.recognizerName> // line 330

+ (void) initialize
{
   [BaseRecognizer setGrammarFileName:@"<fileName>"];
}

+ (NSString *) tokenNameForType:(NSInteger)aTokenType
{
   return [[self getTokenNames] objectAtIndex:aTokenType];
}

+ (<grammar.recognizerName> *)new<grammar.recognizerName>WithCharStream:(id\<CharStream>)anInput
{
   return [[<grammar.recognizerName> alloc] initWithCharStream:anInput];
}

- (id) initWithCharStream:(id\<CharStream>)anInput
{
   self = [super initWithCharStream:anInput State:[RecognizerSharedState
newRecognizerSharedStateWithRuleLen:<numRules>+1]];
   if ( self != nil ) {
<if(memoize)>
       if ( state.ruleMemo == nil ) {
           state.ruleMemo = [[RuleStack newRuleStackWithSize:<numRules>+1] retain];
       }
       if ( [state.ruleMemo count] == 0 ) {
           // initialize the memoization cache
 - the indices are 1-based in the runtime code!
           <! [state.ruleMemo addObject:[NSNull null]];     /* dummy entry to ensure 1-basedness. */ !>
           for (NSInteger i = 0; i \< <numRules>; i++) {
               [state.ruleMemo addObject:[HashRule newHashRuleWithLen:17]];
           }
       }
<endif>
```

```
        <synpreds:{pred | <lexerSynpred(name=pred)>};separator="\n">
        <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = [DFA<dfa.decisionNumber>
newDFA<dfa.decisionNumber>WithRecognizer:self];}; separator="\n">
        <actions.lexer.init>
    }
    return self;
}

- (void) dealloc
{
    <cyclicDFAs:{dfa | [dfa<dfa.decisionNumber> release];}; separator="\n">
<actions.lexer.dealloc>
    [super dealloc];
}

/* ObjC Start of actions.lexer.methods */
<actions.lexer.methods>
/* ObjC end of actions.lexer.methods */
/* ObjC start methods() */
<@methods()>
/* ObjC end methods() */

<if(actions.lexer.reset)>
- (void) reset
{
    <actions.lexer.reset>
    [super reset];
}
<endif>

<if(filterMode)>
<filteringNextToken()>
<endif>
/*
 Start of Rules */
<rules; separator="\n">

@end /* end of <grammar.recognizerName> implementation line 397 */
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
- (id\<Token>) nextToken
```

```
{
   while (YES) {
       if ( [input LA:1] == CharStreamEOF ) {
           return [<labelType> eofToken];
       }
       state.token = nil;
       state.channel = TokenChannelDefault;
       state.tokenStartCharIndex = input.index;
       state.tokenStartCharPositionInLine = input.charPositionInLine;
       state.tokenStartLine = input.line;
       state.text = nil;
       @try {
           NSInteger m = [input mark];
           state.backtracking
 = 1; /* means we won't throw slow exception */
           state.failed = NO;
           [self mTokens];
           state.backtracking = 0;
           /* mTokens backtracks with synpred at backtracking==2
              and we set the synpredgate to allow actions at level 1. */
           if ( state.failed ) {
               [input rewind:m];
               [input consume]; /* advance one char and try again */
           } else {
               [self emit];
               return state.token;
           }
       }
       @catch (RecognitionException *re) {
           // shouldn't happen in backtracking mode, but...
           [self reportError:re];
           [self recover:re];
       }
   }
}

- (void)memoize:(id\<IntStream\>)anInput
     RuleIndex:(NSInteger)ruleIndex
    StartIndex:(NSInteger)ruleStartIndex
{
   if ( state.backtracking > 1 ) [super memoize:anInput RuleIndex:ruleIndex StartIndex:ruleStartIndex];
}

- (BOOL)alreadyParsedRule:(id\<IntStream\>)anInput
 RuleIndex:(NSInteger)ruleIndex
{
   if ( state.backtracking > 1 ) return [super alreadyParsedRule:anInput RuleIndex:ruleIndex];
   return NO;
```

```
}
>>

actionGate() ::= "state.backtracking == 0"

filteringActionGate() ::= "state.backtracking == 1"

parserHeaderFile( LEXER,
          PARSER,
          TREE_PARSER,
          actionScope,
          actions,
          docComment,
          recognizer,
          name,
          tokens,
          tokenNames,
          rules,
          cyclicDFAs,
          bitsets,
          buildTemplate,
          profile,
          backtracking,
          synpreds,
          memoize,
          numRules,
          fileName,
          ANTLRVersion,
          generatedTimestamp,
          trace,
          scopes,
          literals,
          superClass="Parser"
          ) ::= <<
/* parserHeaderFile */
<genericParserHeaderFile(inputStreamType="id\<TokenStream>",...)>
>>

treeParserHeaderFile(
 LEXER,
          PARSER,
          TREE_PARSER,
          actionScope,
          actions,
          docComment,
          recognizer,
          name,
          tokens,
          tokenNames,
          rules,
          cyclicDFAs,
          bitsets,
          buildTemplate,
          profile,
          backtracking,
          synpreds,
          memoize,
          numRules,
          fileName,
          ANTLRVersion,
          generatedTimestamp,
          trace,
          scopes,
          literals,
          superClass="TreeParser"
          ) ::= <<
/* treeParserHeaderFile */
<genericParserHeaderFile(inputStreamType="id\<TreeNodeStream>",...)>
>>

genericParserHeaderFile( LEXER,
          PARSER,
          TREE_PARSER,
          actionScope,
          actions,
          docComment,
          recognizer,
          name,
          tokens,
          tokenNames,
          rules,
          cyclicDFAs,
          bitsets,
           buildTemplate,
          profile,
          backtracking,
          synpreds,
          memoize,
          numRules,
          fileName,
          ANTLRVersion,
          generatedTimestamp,
          trace,
          scopes,
          superClass,
          literals,
        inputStreamType
        ) ::=
<<
<if(actions.(actionScope).header)>
/* =============================================================================
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =============================================================================
*/
<endif>

#ifndef ANTLR3TokenTypeAlreadyDefined
#define ANTLR3TokenTypeAlreadyDefined
typedef enum {
  ANTLR_EOF = -1,
  INVALID,
  EOR,
  DOWN,
  UP,
  MIN
} ANTLR3TokenType;
#endif

<cyclicDFAs:cyclicDFAInterface()>
#pragma mark Tokens
#ifndef TOKENLISTAlreadyDefined
#define TOKENLISTAlreadyDefined
 1
#ifdef EOF
#undef EOF
#endif
<tokens:{it | #define <it.name> <it.type>}; separator="\n">
#endif
#pragma mark Dynamic Global Scopes globalAttributeScopeInterface
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeInterface(scope=it)><endif>}>
#pragma mark Dynamic Rule Scopes ruleAttributeScopeInterface
<rules:{rule |
<rule.ruleDescriptor:{ ruleDescriptor | <ruleAttributeScopeInterface(scope=ruleDescriptor.ruleScope)>}>}>
#pragma mark Rule Return Scopes returnScopeInterface
<rules:{rule |<rule.ruleDescriptor:{ ruleDescriptor |
<returnScopeInterface(scope=ruleDescriptor.returnScope)>}>}>

/* Interface grammar class */
@interface <className()> <@superClassName> : <superClass><@end> { /* line 572 */
#pragma mark Dynamic Rule Scopes ruleAttributeScopeDecl
<rules:{rule | <rule.ruleDescriptor.ruleScope:ruleAttributeScopeDecl(scope=rule.ruleDescriptor.ruleScope)>}>
```

```
#pragma mark Dynamic Global Rule Scopes globalAttributeScopeMemVar
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeMemVar(scope=it)><endif>}><\n>
/*
 ObjC start of actions.(actionScope).memVars */
<actions.(actionScope).memVars>
/* ObjC end of actions.(actionScope).memVars */
/* ObjC start of memVars */
<@memVars()>
/* ObjC end of memVars */

<cyclicDFAs:{dfa | DFA<dfa.decisionNumber> *dfa<dfa.decisionNumber>;}; separator="\n">
<synpreds:{pred | SEL <pred>Selector;}; separator="\n">
}

/* ObjC start of actions.(actionScope).properties */
<actions.(actionScope).properties>
/* ObjC end of actions.(actionScope).properties */
/* ObjC start of properties */
<@properties()>
/* ObjC end of properties */

+ (void) initialize;
+ (<className()> *) new<className()>:(<inputStreamType>)aStream;
/* ObjC start of actions.(actionScope).methodsDecl */
<actions.(actionScope).methodsDecl>
/* ObjC end of actions.(actionScope).methodsDecl */

/* ObjC start of methodsDecl */
<@methodsDecl()>
/* ObjC end of methodsDecl */

<rules:{rule |
-
(<rule.ruleDescriptor:{ruleDescriptor|<returnType()>}>)<if(!rule.ruleDescriptor.isSynPred)><rule.ruleName><else
><rule.ruleName>_fragment<endif><if(rule.ruleDescriptor.parameterScope)><rule.ruleDescriptor.parameterScope:
parameterScope()><endif>;
 }; separator="\n"><\n>

@end /* end of <className()> interface */<\n>
>>

parser( grammar,
      name,
      scopes,
      tokens,
      tokenNames,
      rules,
      numRules,
      bitsets,
      ASTLabelType="CommonTree",
      superClass="Parser",
      labelType="CommonToken",
      members={<actions.parser.members>}
      ) ::= <<
<genericParser(inputStreamType="id\<TokenStream>", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser( grammar,
      name,
      scopes,
      tokens,
      tokenNames,
      globalAction,
      rules,
      numRules,
      bitsets,
      filterMode,
      labelType={<ASTLabelType>},
      ASTLabelType="CommonTree",
       superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
      members={<actions.treeparser.members>}
      ) ::= <<
<genericParser(inputStreamType="id\<TreeNodeStream>", rewriteElementType="Node", ...)>
>>

/** How to generate a parser */
genericParser(  grammar,
      name,
      scopes,
      tokens,
      tokenNames,
      rules,
      numRules,
      cyclicDFAs,         // parser init -- initializes the DFAs
      bitsets,
      labelType,
      ASTLabelType,
      superClass,
      members,
      filterMode,
      rewriteElementType,
      inputStreamType
      ) ::= <<
```

```
<cyclicDFAs:cyclicDFA()>

#pragma mark Bitsets
<bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>}>

#pragma mark Dynamic Global globalAttributeScopeImplementation
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeImplementation(scope=it)><endif>}>

#pragma mark Dynamic
 Rule Scopes ruleAttributeScopeImplementation
<rules:{rule |
<rule.ruleDescriptor:{ ruleDescriptor | <ruleAttributeScopeImplementation(scope=ruleDescriptor.ruleScope)>}>}>

#pragma mark Rule Return Scopes returnScopeImplementation
<rules:{rule | <rule.ruleDescriptor:{ ruleDescriptor |
<returnScopeImplementation(scope=ruleDescriptor.returnScope)>}>}>

@implementation <grammar.recognizerName>  // line 637

/* ObjC start of ruleAttributeScope */
#pragma mark Dynamic Rule Scopes ruleAttributeScope
<rules:{rule | <rule.ruleDescriptor.ruleScope:ruleAttributeScope()>}>
/* ObjC end of ruleAttributeScope */
#pragma mark global Attribute Scopes globalAttributeScope
/* ObjC start globalAttributeScope */
<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScope()><endif>}>
/* ObjC end globalAttributeScope */
/* ObjC start actions.(actionScope).synthesize */
<actions.(actionScope).synthesize>
/* ObjC start synthesize() */
<@synthesize()>

+ (void) initialize
{
   #pragma mark Bitsets

  <bitsets:{it | <bitsetInit(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>}>
   [BaseRecognizer setTokenNames:[[AMutableArray arrayWithObjects:@"\<invalid>", @"\<EOR>",
@"\<DOWN>", @"\<UP>", <tokenNames:{it | @<it>}; separator=", ", wrap="\n ">, nil] retain]];
   [BaseRecognizer setGrammarFileName:@"<fileName>"];
   <synpreds:{pred | <synpred(pred)>}>
}

+ (<grammar.recognizerName> *)new<grammar.recognizerName>:(<inputStreamType>)aStream
{
<if(PARSER)>
   return [[<grammar.recognizerName> alloc] initWithTokenStream:aStream];
<else><! TREE_PARSER !>
```

```
  return [[<grammar.recognizerName> alloc] initWithStream:aStream];
<endif>
}

<if(PARSER)>
- (id) initWithTokenStream:(<inputStreamType>)aStream
{
  self = [super initWithTokenStream:aStream State:[[RecognizerSharedState
newRecognizerSharedStateWithRuleLen:<numRules>+1] retain]];
  if ( self != nil ) {
<else><! TREE_PARSER !>
- (id) initWithStream:(<inputStreamType>)aStream
{
  self = [super initWithStream:aStream
 State:[[RecognizerSharedState newRecognizerSharedStateWithRuleLen:<numRules>+1] retain]];
  if ( self != nil ) {
<endif>
      <! <parserCtorBody()> !>
      <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = [DFA<dfa.decisionNumber>
newDFA<dfa.decisionNumber>WithRecognizer:self];}; separator="\n">
      <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeInit(scope=it)><endif>}>
      <rules:{rule | <rule.ruleDescriptor.ruleScope:ruleAttributeScopeInit()>}>
      /* start of actions-actionScope-init */
      <actions.(actionScope).init>
      /* start of init */
      <@init()>
  }
  return self;
}

- (void) dealloc
{
  <cyclicDFAs:{dfa | [dfa<dfa.decisionNumber> release];}; separator="\n">
  <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeDealloc(scope=it)><endif>}>
  <actions.(actionScope).dealloc>
  <@dealloc()>
  [super dealloc];
}

/* ObjC start actions.(actionScope).methods */
<actions.(actionScope).methods>
/* ObjC
 end actions.(actionScope).methods */
/* ObjC start methods() */
<@methods()>
/* ObjC end methods() */
/* ObjC start rules */
<rules; separator="\n">
```

```
/* ObjC end rules */

@end /* end of <grammar.recognizerName> implementation line 692 */<\n>
>>

parserCtorBody() ::= <<
<if(memoize)> /* parserCtorBody */
<if(grammar.grammarIsRoot)>
state.ruleMemo = [[RuleStack newRuleStack:<numRules>+1] retain];<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock)
 ::=
<<
// $ANTLR start <ruleName>_fragment
- (void) <ruleName>_fragment
{
   <ruleLabelDefs()>
   <if(trace)>
       [self traceIn:\@"<ruleName>_fragment" Index:<ruleDescriptor.index>];
   @try {
       <block>
   }
   @finally {
       [self traceOut:\@"<ruleName>_fragment" Index:<ruleDescriptor.index>];
   }
<else>
   <block>
<endif>
} // $ANTLR end <ruleName>_fragment
>>

synpred(name) ::= <<
SEL <name>Selector = @selector(<name>_fragment);
<! // $ANTLR start <name>
- (BOOL) <name>
{
   state.backtracking++;
   <@start()>
   NSInteger start = [input mark];
   @try {
       [self <name>_fragment]; // can never throw exception
   }
   @catch (RecognitionException *re) {
       NSLog(@"impossible: %@\n", re.name);
   }
   BOOL success = (state.failed == NO);
   [input rewind:start];
   <@stop()>
   state.backtracking--;
   state.failed=NO;
   return success;
} // $ANTLR end <name> <\n> !>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name)
 ::= <<
<if(memoize)>
if ( state.backtracking > 0 && [self alreadyParsedRule:input RuleIndex:<ruleDescriptor.index>] ) { return
<ruleReturnValue()>; }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if (backtracking)>if ( state.failed ) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if ( state.backtracking > 0 ) { state.failed = YES; return <ruleReturnValue()>; }<\n><endif>
>>

/** How to generate code for a rule.
*  The return type aggregates are declared in the header file (headerFile template)
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<

/*
* $ANTLR start <ruleName>
* <fileName>:<description>
```

```
*/
- (<returnType()>) <ruleName><ruleDescriptor.parameterScope:parameterScope()>
{
   <if(trace)>[self traceIn:\@"<ruleName>" Index:<ruleDescriptor.index>];<endif>
   <if(trace)>NSLog(@"enter
 <ruleName> %@ failed=%@ backtracking=%d", [input LT:1], (state.failed==YES)?@"YES":@"NO",
state.backtracking);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   @try {
       <ruleMemoization(name=ruleName)>
       <ruleLabelDefs()>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
   }
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else><if(!emptyRule)><if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   @catch (RecognitionException *re) {
       [self reportError:re];
       [self recover:input Exception:re];
       <@setErrorReturnValue()>
   }<\n>
<endif><endif><endif>
   @finally {
       <if(trace)>[self traceOut:@"<ruleName>" Index:<ruleDescriptor.index>];<endif>
       <memoize()>
       <ruleScopeCleanUp()>
       <finally>
   }
   <@postamble()>
   return <ruleReturnValue()>;
}
/*
 $ANTLR end <ruleName> */
>>

finalCode(finalBlock) ::= <<
{
   <finalBlock>
}
>>
```

```
catch(decl,action) ::= <<
@catch (<e.decl>) {
   <e.action>
}
>>

ruleDeclarations() ::= <<
/* ruleDeclarations */
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = [<ruleDescriptor:returnStructName()> new<ruleDescriptor:returnStructName()>];
[retval setStart:[input LT:1]];<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
NSInteger <ruleDescriptor.name>_StartIndex = input.index;
<endif>
>>

ruleScopeSetUp() ::= <<
/* ruleScopeSetUp */
<ruleDescriptor.useScopes:{it | [<it>_stack push:[<it>_Scope new<it>_Scope]];}>
<ruleDescriptor.ruleScope:{it | [<it.name>_stack push:[<it.name>_Scope new<it.name>_Scope]];}>
>>

ruleScopeCleanUp() ::= <<
/* ruleScopeCleanUp */
<ruleDescriptor.useScopes:{it | [<it>_stack pop];}; separator="\n">
<ruleDescriptor.ruleScope:{it | [<it.name>_stack
 pop];}; separator="\n">
>>

ruleLabelDefs() ::= <%
/* ruleLabelDefs entry */<"\n">
<[ruleDescriptor.tokenLabels, ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it |<labelType> *<it.label.text> = nil;}; separator="\n"><"\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it | AMutableArray *list_<it.label.text> = nil;}; separator="\n"><"\n">
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n"><"\n">
<ruleDescriptor.ruleListLabels:{ll|ParserRuleReturnScope *<ll.label.text> = nil;}; separator="\n"><"\n">
%>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |<labelType> *<it.label.text>=nil;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |NSInteger <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]:{it |AMutableArray
 *list_<it.label.text>=nil; }; separator="\n">
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanUp() ::= <<
/* token+rule list labels */
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{it |[list_<it.label.text> release];}; separator="\n">
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
[retval setStop:[input LT:-1]];<\n>
<endif><endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (state.backtracking > 0) [self memoize:input RuleIndex:<ruleDescriptor.index>
StartIndex:<ruleDescriptor.name>_StartIndex];
<endif><endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName, nakedBlock, ruleDescriptor, block, memoize) ::= <<
// $ANTLR start "<ruleName>"
- (void)
m<ruleName><if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope:parameterScope(scope=it)><en
dif>
{
   //<if(trace)>[self traceIn:\@"<ruleName>" Index:<ruleDescriptor.index>];<endif>
   <if(trace)>NSLog(@"enter <ruleName> %C line=%d:%d failed=%@ backtracking=%d",
       [input LA:1],
       self.line,
       self.charPositionInLine,
       (state.failed==YES) ? @"YES" : @"NO",
       state.backtracking);
   <endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   @try {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       NSInteger _type = <ruleName>;
       NSInteger _channel = TokenChannelDefault;
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       state.type = _type;
       state.channel = _channel;
       <(ruleDescriptor.actions.after):execAction()>
<endif>
   }
   @finally {

 //<if(trace)>[self traceOut:[NSString stringWithFormat:@"<ruleName> %d\n", <ruleDescriptor.index>]];<endif>
       <if(trace)>NSLog(@"exit <ruleName> %C line=%d:%d failed=%@ backtracking=%d",
               [input LA:1], self.line, self.charPositionInLine,
               (state.failed==YES) ? @"YES" : @"NO", state.backtracking);<endif>
       <ruleScopeCleanUp()>
       <memoize()>
   }
   return;
}
/* $ANTLR end "<ruleName>" */
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
- (void) mTokens
```

```
{
   <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description> // block
NSInteger alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
 {
   <alts:{a | <altSwitchCase(i, a)>}>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description> //ruleblock
NSInteger alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
   <alts:{a | <altSwitchCase(i, a)>}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description> // ruleBlockSingleAlt
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description> // blockSingleAlt
<decls>
```

```
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n)
 ::= <<
// <fileName>:<description> // positiveClosureBlock
NSInteger cnt<decisionNumber> = 0;
<decls>
<@preloop()>
do {
   NSInteger alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
       <alts:{a | <altSwitchCase(i, a)>}>
       default :
           if ( cnt<decisionNumber> >= 1 )
               goto loop<decisionNumber>;
           <ruleBacktrackFailure()>
           EarlyExitException *eee =
               [EarlyExitException newException:input decisionNumber:<decisionNumber>];
           <@earlyExitException()>
           @throw eee;
   }
   cnt<decisionNumber>++;
} while (YES);
loop<decisionNumber>: ;
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 0 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
 <<
<decls>
<@preloop()>
do {
   NSInteger alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
    <alts:{a | <altSwitchCase(i, a)>}>
    default :
      goto loop<decisionNumber>;
  }
} while (YES);
loop<decisionNumber>: ;
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum, alt) ::= <<
case <altNum> : ;
  <@prealt()>
  <alt>
  break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew)
 ::= <<
// <fileName>:<description> // alt
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S
```

```
/** Dump the elements one per line */
element(e) ::= << <@prematch()><\n><e.el><\n> >>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label>=(<labelType> *)<endif>[self match:input TokenType:<token>
Follow:FOLLOW_<token>_in_<ruleName><elementIndex>]; <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == nil) list_<label> = [[AMutableArray arrayWithCapacity:5] retain];
[list_<label> addObject:<elem>];<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>NSInteger <label> =
 [input LA:1];<\n><endif>
[self matchChar:<char>]; <checkRuleBacktrackFailure()><\n>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)><label> = [input LA:1];<\n><endif>
[self matchRangeFromChar:<a> to:<b>]; <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label> = [input LA:1];<\n>
<else>
<label> = (<labelType> *)[input LT:1]; /* matchSet */<\n>
<endif><endif>
if (<s>) {
   [input consume];
   <postmatchCode>
<if(!LEXER)>
   [state setIsErrorRecovery:NO];
<endif>
   <if(backtracking)>state.failed = NO;<\n><endif>
} else {
   <ruleBacktrackFailure()>
   MismatchedSetException *mse = [MismatchedSetException newException:nil stream:input];
   <@mismatchedSetException()>
<if(LEXER)>
<if(label)>
   mse.c = <label>;
<endif>
   [self recover:mse];
   @throw mse;
<else>
   @throw mse;
   <! use following code to
 make it recover inline; remove throw mse;
   [self recoverFromMismatchedSet:input exception:mse follow:FOLLOW_set_in_<ruleName><elementIndex>]; !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
NSInteger <label>Start = input.index;
[self matchString:<string>]; <checkRuleBacktrackFailure()>
NSInteger StartLine<elementIndex> = self.line;
NSInteger <label>StartCharPos<elementIndex> = self.charPositionInLine;
<label> = [[<labelType> newToken:input Type:TokenTypeInvalid Channel:TokenChannelDefault
Start:<label>Start Stop:input.index] retain];
[self setLine:<label>StartLine<elementIndex>];
[self setCharPositionInLine:<label>StartCharPos<elementIndex>];
<else>
[self matchString:<string>]; <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions)
 ::= <<
<if(label)>
<label> = (<labelType> *)[input LT:1];<\n>
<endif>
[self matchAny:input]; <checkRuleBacktrackFailure()>
```

```
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
NSInteger <label> = [input LA:1];<\n>
<endif>
[self matchAny]; <checkRuleBacktrackFailure()><\n>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
/* ruleRef */
[self pushFollow:FOLLOW_<rule.name>_in_<ruleName><elementIndex>];
<if(label)><label> = <endif>[self <if(scope)><scope:delegateName()>.<endif><rule.name><if(args)>:<first(args)>
 <rest(args):{ a | arg<i>:<rest(args)>}; separator=" "><endif>];<\n>
[self popFollow];
<checkRuleBacktrackFailure()><\n>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
```

```
<if(label)>
NSInteger <label>Start<elementIndex> = input.index;
[self m<rule.name><if(args)>:<args; separator=" :"><endif>]; <checkRuleBacktrackFailure()><\n>
<label> = [[<labelType> newToken:input Type:TokenTypeInvalid Channel:TokenChannelDefault
Start:<label>Start<elementIndex> Stop:input.index-1] retain];
<label>.line = self.line;
<else>
[self <if(scope)><scope:delegateName()>.<endif>m<rule.name><if(args)>:<args; separator="
 :"><endif>]; <checkRuleBacktrackFailure()><\n>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
NSInteger <label>Start<elementIndex> = input.index;
[self matchChar:CharStreamEOF]; <checkRuleBacktrackFailure()><\n>
<labelType> <label> = [[<labelType> newToken:input Type:TokenTypeEOF Channel:TokenChannelDefault
Start:<label>Start<elementIndex> Stop:input.index-1] retain];
<label>.line = self.line;
<else>
[self matchChar:CharStreamEOF]; <checkRuleBacktrackFailure()><\n>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()                       ::= "int <recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::=
 "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( [input LA:1] == DOWN ) {
   [self match:input TokenType:DOWN Follow:nil]; <checkRuleBacktrackFailure()>
   <children:element()>
   [self match:input TokenType:UP Follow:nil]; <checkRuleBacktrackFailure()>
}
<else>
   [self match:input TokenType:DOWN Follow:nil]; <checkRuleBacktrackFailure()>
   <children:element()>
   [self match:input TokenType:UP Follow:nil]; <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
   <ruleBacktrackFailure()>
   @throw [FailedPredicateException newException:@"<ruleName>" predicate:@"<description>" stream:input];
}
>>

//
 Fixed  DFA  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:<k>];<\n>
<edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   NoViableAltException *nvae = [NoViableAltException newException:<decisionNumber> state:<stateNumber>
stream:input];
   nvae.c = LA<decisionNumber>_<stateNumber>;
   <@noViableAltException()>
   @throw nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:<k>];<\n>
<edges; separator="\nelse ">
>>
```

```
/**
 A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:<k>];
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
   alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif><endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates
 for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
unichar charLA<decisionNumber> = [input LA:<k>];
switch (charLA<decisionNumber>) {
   <edges; separator="\n"><\n>
default: ;
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
```

```
  NoViableAltException *nvae = [NoViableAltException newException:<decisionNumber> state:<stateNumber>
stream:input];
  nvae.c = charLA<decisionNumber>;
  <@noViableAltException()>
  @throw nvae;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>])
 { // dfaOptionalBlockStateSwitch
  <edges; separator="\n"><\n>
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>]) { // dfaLoopbackStateSwitch
  <edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
  alt<decisionNumber> = <eotPredictsAlt>;
  break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it | case <it>: ;}; separator="\n">
  {
  <targetState>
  }
  break;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = [dfa<decisionNumber> predict:input];
>>

/** Used in headerFile */
cyclicDFAInterface(dfa) ::= <<
#pragma mark Cyclic DFA interface start DFA<dfa.decisionNumber>
```

```
@interface DFA<dfa.decisionNumber> : DFA {
}
+ (DFA<dfa.decisionNumber>
 *) newDFA<dfa.decisionNumber>WithRecognizer:(BaseRecognizer *)theRecognizer;
- initWithRecognizer:(BaseRecognizer *)recognizer;
@end /* end of DFA<dfa.decisionNumber> interface  */<\n>
#pragma mark Cyclic DFA interface end DFA<dfa.decisionNumber><\n>
>>

/** Used in lexer/parser implementation files */
/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
#pragma mark Cyclic DFA implementation start DFA<dfa.decisionNumber>

@implementation DFA<dfa.decisionNumber>
const static NSInteger dfa<dfa.decisionNumber>_eot[<dfa.numberOfStates>] =
   {<dfa.eot; wrap="\n     ", separator=",", null="-1">};
const
 static NSInteger dfa<dfa.decisionNumber>_eof[<dfa.numberOfStates>] =
   {<dfa.eof; wrap="\n     ", separator=",", null="-1">};
const static unichar dfa<dfa.decisionNumber>_min[<dfa.numberOfStates>] =
   {<dfa.min; wrap="\n     ", separator=",", null="-1">};
const static unichar dfa<dfa.decisionNumber>_max[<dfa.numberOfStates>] =
   {<dfa.max; wrap="\n     ", separator=",", null="-1">};
const static NSInteger dfa<dfa.decisionNumber>_accept[<dfa.numberOfStates>] =
   {<dfa.accept; wrap="\n     ", separator=",", null="-1">};
const static NSInteger dfa<dfa.decisionNumber>_special[<dfa.numberOfStates>] =
   {<dfa.special; wrap="\n     ", separator=",", null="-1">};

/** Used when there is no transition table entry for a particular state */
#define dfa<dfa.decisionNumber>_T_empty     nil

<dfa.edgeTransitionClassMap.keys:{ table |
const static NSInteger dfa<dfa.decisionNumber>_T<i0>[] =
{
    <table; separator=", ", wrap="\n ", null="-1">
\};
}; null="">

const static NSInteger *dfa<dfa.decisionNumber>_transition[]
```

```
 =
{
   <dfa.transitionEdgeTables:{whichTable|dfa<dfa.decisionNumber>_T<whichTable>}; separator=", ", wrap="\n",
null="nil">
};

+ (DFA<dfa.decisionNumber> *) newDFA<dfa.decisionNumber>WithRecognizer:(BaseRecognizer *)aRecognizer
{
   return [[[DFA<dfa.decisionNumber> alloc] initWithRecognizer:aRecognizer] retain];
}

- (id) initWithRecognizer:(BaseRecognizer *) theRecognizer
{
   self = [super initWithRecognizer:theRecognizer];
   if ( self != nil ) {
       decisionNumber = <dfa.decisionNumber>;
       eot = dfa<dfa.decisionNumber>_eot;
       eof = dfa<dfa.decisionNumber>_eof;
       min = dfa<dfa.decisionNumber>_min;
       max = dfa<dfa.decisionNumber>_max;
       accept = dfa<dfa.decisionNumber>_accept;
       special = dfa<dfa.decisionNumber>_special;
       transition = dfa<dfa.decisionNumber>_transition;
   }
   return self;
}

<if(dfa.specialStateSTs)>
/* start dfa.specialStateSTs */
- (NSInteger) specialStateTransition:(NSInteger)s
 Stream:(id\<IntStream\>)anInput
{
<if(LEXER)>
   id\<IntStream\> input = anInput;<\n>
<endif>
<if(PARSER)>
   id\<TokenStream\> input = (id\<TokenStream\>)anInput;<\n>
<endif>
<if(TREE_PARSER)>
   id\<TreeNodeStream\> input = (id\<TreeNodeStream\>)anInput;<\n>
<endif>
   switch (s) {
       <dfa.specialStateSTs:{state |
       case <i0> : ;<! compressed special state numbers 0..n-1 !>
           <state>}; separator="\n">
   }
<if(backtracking)>
   if ( [recognizer getBacktrackingLevel] > 0 ) { [recognizer setFailed:YES]; return -1; }<\n>
<endif>
   NoViableAltException *nvae = [NoViableAltException newException:<dfa.decisionNumber> state:s
stream:recognizer.input];
   // nvae.c = s;
   /* [self error:nvae]; */ <! for debugger - do later !>
   @throw nvae;
}<\n>
/* end dfa.specialStateSTs */
<endif>

- (void) dealloc
{
   //free(transition);
   [super dealloc];
}

- (NSString *) description
{
   return @"<dfa.description>";
}

<@errorMethod()>

@end /* end DFA<dfa.decisionNumber>
 implementation */<\n>
#pragma mark Cyclic DFA implementation end DFA<dfa.decisionNumber>
<\n>
>>
/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber, stateNumber, edges, needErrorClause, semPredState) ::= <<
/* cyclicDFAState */
NSInteger LA<decisionNumber>_<stateNumber> = [input LA:1];<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
NSInteger index<decisionNumber>_<stateNumber> = input.index;
[input rewind];<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
[input seek:index<decisionNumber>_<stateNumber>];<\n>
<endif>
if ( s >= 0 )
   return s;
break;
>>
```

```
/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
/* cyclicDFAEdge */
if (<labelExpr><if(predicates)> && (<predicates>)<endif>)
 { s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n> /* eotDFAEdge */
>>

// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(pred, {})>)"

evalPredicate(pred,description) ::= "(<pred>)"

/*
* evalSynPredicate(pred,description) ::= "<pred>()"
*
* synpreds are broken in cyclic DFA special states
*  Damn! For now, work around with using the selectors directly, and by providing a trampoline evalSynPred
method in
*  DFA
*/
/* evalSynPredicate(pred,description) ::= "[self evaluateSyntacticPredicate:<pred>Selector stream:input]" */
evalSynPredicate(pred,description) ::= "[self evaluateSyntacticPredicate:@selector(<pred>_fragment)]"
/* evalSynPredicate(pred,description)
 ::= "[recognizer <pred>]" */

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "[input LA:<k>] == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<stateNumber> >= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
```

```
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(([input LA:<k>] >=
<lower>) && ([input LA:<k>] \<= <upper>))"

setTest(ranges) ::= <%
<ranges; separator="||">
%>

// A T T R I B U T E S

memVars(scope) ::= <% <scope.attributes:{a|<a.type> <a.name>;<\n>}; separator="\n"> %>

properties(scope) ::= <%
<scope.attributes:{a|@property (assign, getter=get<a.name>, setter=set<a.name>:) <a.type> <a.name>;<\n>};
separator="\n">
%>

methodsDecl(scope) ::= <%
<scope.attributes:{a|-
 (<a.type>)get<a.name>;<\n>- (void)set<a.name>:(<a.type>)aVal;<\n>}; separator="\n">
%>

synthesize(scope) ::= <% <scope.attributes:{a|@synthesize <a.name>;}; separator="\n"> %>

methods(scope) ::= <%
<scope.attributes:{a|
- (<a.type>)get<a.name> { return( <a.name> ); \}<\n>
- (void)set<a.name>:(<a.type>)aVal { <a.name> = aVal; \}<\n>}; separator="\n">
%>

globalAttributeScopeInterface(scope) ::= <%
/* globalAttributeScopeInterface */<\n>
@interface <scope.name>_Scope : SymbolsScope {<\n>
<if(scope.attributes)>
<memVars(scope)>
<endif>
}<\n>
<if(scope.attributes)>
/* start of globalAttributeScopeInterface properties */<\n>
<properties(scope)>
/* end globalAttributeScopeInterface properties */<\n>
<endif>

+ (<scope.name>_Scope *)new<scope.name>_Scope;<\n>
- (id) init;<\n>
<if(scope.attributes)>
/* start of globalAttributeScopeInterface methodsDecl */<\n>
```

```
<methodsDecl(scope)>
/* End of globalAttributeScopeInterface methodsDecl */<\n>
<endif>
@end /* end of <scope.name>_Scope interface
 */<\n>
%>

globalAttributeScopeMemVar(scope) ::= <%
/* globalAttributeScopeMemVar */<\n>
SymbolStack *<scope.name>_stack;<\n>
<scope.name>_Scope *<scope.name>_scope;<\n>
%>

globalAttributeScopeImplementation(scope) ::= <%
@implementation <scope.name>_Scope  /* globalAttributeScopeImplementation */<\n>
<if(scope.attributes)>
/* start of synthesize -- OBJC-Line 1750 */<\n>
<synthesize(scope)><\n>
<endif>
<\n>
+ (<scope.name>_Scope *)new<scope.name>_Scope<\n>
{<\n>
   return [[<scope.name>_Scope alloc] init];<\n>
}<\n>
<\n>
- (id) init<\n>
{<\n>
   self = [super init];<\n>
   return self;<\n>
}<\n>
<\n>
<if(scope.attributes)>
/* start of iterate get and set functions */<\n>
<methods(scope)><\n>
/* End of iterate get and set functions */<\n>
<endif>
@end /* end of <scope.name>_Scope implementation */<\n><\n>
%>

globalAttributeScopeInit(scope) ::= <<
/* globalAttributeScopeInit */<\n>
<scope.name>_scope = [<scope.name>_Scope new<scope.name>_Scope];<\n>
<scope.name>_stack = [SymbolStack
 newSymbolStackWithLen:30];<\n>
>>

globalAttributeScopeDealloc(scope) ::= <% [<scope.name>_stack release];<\n> %>
```

```
globalAttributeScope(scope) ::= <%
<if(scope.name)>
static <scope.name>_stack;<\n>
<endif>
%>

ruleAttributeScopeMemVar(scope) ::= <%
/* ObjC ruleAttributeScopeMemVar */<\n>
<if(scope.attributes)>
<scope.name>_Scope *<scope.name>_scope; /* ObjC ruleAttributeScopeMemVar */<\n>
<endif>
%>

ruleAttributeScopeInterface(scope) ::= <%
<if(scope.attributes)>
/* start of ruleAttributeScopeInterface */<\n>
@interface <scope.name>_Scope : SymbolsScope {<\n>
   <memVars(scope)><\n>
}<\n>
<\n>
/* start property declarations */<\n>
<properties(scope)><\n>
/* start method declarations */<\n>
+ (<scope.name>_Scope *)new<scope.name>_Scope;<\n>
- (id) init;<\n>
<methodsDecl(scope)><\n>
@end /* end of ruleAttributeScopeInterface */<\n><\n>
<endif>
%>

ruleAttributeScopeImplementation(scope) ::= <%
<if(scope.attributes)>
@implementation <scope.name>_Scope  /* start of ruleAttributeScopeImplementation
 */<\n>
<synthesize(scope)><\n>
<\n>
+ (<scope.name>_Scope *)new<scope.name>_Scope<\n>
{<\n>
   return [[<scope.name>_Scope alloc] init];<\n>
}<\n>
<\n>
- (id) init<\n>
{<\n>
   self = [super init];<\n>
   return self;<\n>
}<\n>
<\n>
/* start of <scope.name>_Scope get and set functions */<\n>
```

```
<methods(scope)><\n>
/* End of <scope.name>_Scope get and set functions */<\n>
@end /* end of ruleAttributeScopeImplementation */<\n><\n>
<endif>
%>

ruleAttributeScopeInit(scope) ::= <%
/* ruleAttributeScopeInit */<\n>
<scope.name>_scope = [<scope.name>_Scope new<scope.name>_Scope];<\n>
<scope.name>_stack = [SymbolStack newSymbolStackWithLen:30];<\n>
%>

ruleAttributeScopeDealloc(scope) ::= <% [<scope.name>_Scope release];<\n> %>

ruleAttributeScope(scope) ::= <%
<if(scope.attributes)>
/* ruleAttributeScope */<\n>
static SymbolStack *<scope.name>_stack;<\n>
static <scope.name>_Scope *<scope.name>_scope;
<endif>
%>

ruleAttributeScopeDecl(scope) ::= <%
/* ruleAttributeScopeDecl
 */<\n>
<if(scope.attributes)>
<scope.name>_Scope *<scope.name>_scope;<\n>
<endif>
%>

returnStructName(r) ::= "<className()>_<r.name>_return"

returnType() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()> *
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
void
<endif>
%>
```

```
/** Generate the Objective-C type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<className()>_<referencedRule.name>_return *<else>
<if(referencedRule.hasSingleReturnValue)><referencedRule.singleValueReturnType><else>
void<endif>
<endif>
%>

delegateName(d) ::= <% <if(d.label)><d.label><else>g<d.name><endif> %>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <% <objcTypeInitMap.(typeName)>
 %>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
%>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScopeInterface(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
/* returnScopeInterface <ruleDescriptor:returnStructName()> */
@interface <ruleDescriptor:returnStructName()> :
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope { /* returnScopeInterface line 1838 */
<@memVars()> /* ObjC start of memVars() */<\n>
<if(scope.attributes)>
<memVars(scope)><\n>
<endif>
}
/* start property declarations */
<@properties()><\n>
<if(scope.attributes)>
<properties(scope)><\n>
<endif>
/* start of method declarations */<\n>
+ (<ruleDescriptor:returnStructName()> *)new<ruleDescriptor:returnStructName()>;
/*
```

```
 this is start of set and get methods */
<@methodsDecl()>  /* methodsDecl */<\n>
<if(scope.attributes)>
/* start of iterated get and set functions */<\n>
<methodsDecl(scope)><\n>
<endif>
@end /* end of returnScopeInterface interface */<\n>
<endif>
>>

returnScopeImplementation(scope) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
@implementation <ruleDescriptor:returnStructName()> /* returnScopeImplementation */<\n>
<@synthesize()> /* start of synthesize -- OBJC-Line 1837 */<\n>
<if(scope.attributes)>
   <synthesize(scope)><\n>
<endif>
+ (<ruleDescriptor:returnStructName()> *)new<ruleDescriptor:returnStructName()><\n>
{<\n>
   return [[[<ruleDescriptor:returnStructName()> alloc] init] retain];<\n>
}<\n>
<\n>
- (id) init<\n>
{<\n>
   self = [super init];<\n>
   return self;<\n>
}<\n>
<\n>
<@methods()><\n>
<if(scope.attributes)>
/* start of iterate get and set functions */<\n>
<methods(scope)><\n>
/* End of iterate get and set functions
 */<\n>
<endif>
<actions.(actionScope).ruleReturnMethods>
<@ruleReturnMembers()><\n>
@end /* end of returnScope implementation */<\n><\n>
<endif>
%>

parameterScope(scope) ::= <<
<! <scope.attributes:{it | :(<it.type>)<it.name>}; separator=" "> !>
<first(scope.attributes):{ a | :(<a.type>)<a.name>}> <rest(scope.attributes):{ a | arg<i>:(<a.type>)<a.name> };
separator=" ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
```

```
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>;"

/** Note that the scopeAttributeRef does not have access to the
* grammar name directly
*/
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
([((<scope>_Scope *)[<scope>_stack objectAtIndex:[<scope>_stack size]-<negIndex>-1)]).<attr.name>
<else>
<if(index)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:<index>]).<attr.name>
<else>
((<scope>_Scope *)[<scope>_stack peek]).<attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
/* scopeSetAttributeRef */
<if(negIndex)>
((<scope>_Scope
 *)[<scope>_stack objectAtIndex:([<scope>_stack size]-<negIndex>-1)]).<attr.name> = <expr>;
<else>
<if(index)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:<index>]).<attr.name> = <expr>;
<else>
((<scope>_Scope *)[<scope>_stack peek]).<attr.name> = <expr>;
<endif>
<endif>
%>

scopeAttributeRefStack() ::= <<
/* scopeAttributeRefStack */
<if(negIndex)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:[<scope>_stack count]-<negIndex>-1]).<attr.name> = <expr>;
<else>
<if(index)>
((<scope>_Scope *)[<scope>_stack objectAtIndex:<index>]).<attr.name> = <expr>;
<else>
((<scope>_Scope *)[<scope>_stack peek]).<attr.name> = <expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
```

```
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return
 value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=nil?<scope>.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>  /* added to returnAttributeRef */<\n>
<else>
<attr.name><\n>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>; /* added to returnSetAttributeRef */<\n>
<else>
<attr.name> = <expr>;<\n>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



/* not sure the next are the right approach; and they are evaluated early; */
/* they cannot see TREE_PARSER or PARSER attributes for example. :( */

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=nil?<scope>.text:nil)"
tokenLabelPropertyRef_type(scope,attr)
 ::= "(<scope>!=nil?<scope>.type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=nil?<scope>.line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=nil?<scope>.charPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=nil?<scope>.channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=nil?[<scope> getTokenIndex]:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=nil?[<scope>.text integerValue]:0)"
```

```
ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=nil?((<labelType> *)<scope>.start):nil)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=nil?((<labelType> *)<scope>.stopToken):nil)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>!=nil?((<ASTLabelType> *)<scope>.tree):nil)"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
(<scope>!=nil?[[input getTokenStream] toStringFromStart:[[input getTreeAdaptor] getTokenStartIndex:[<scope>
getStart]]

   ToEnd:[[input getTreeAdaptor] getTokenStopIndex:[<scope> getStart]]]:0)
<else>
(<scope>!=nil?([input toStringFromToken:[<scope> getStart] ToToken:[<scope> getStop]]:0)
<endif>
%>
ruleLabelPropertyRef_st(scope,attr) ::= "(<scope>!=nil?[<scope> st]:nil)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(<scope>!=nil?<scope>.type:0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(<scope>!=nil?<scope>.line:0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=nil?<scope>.charPositionInLine:-1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=nil?<scope>.channel:0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(<scope>!=nil?[<scope> getTokenIndex]:0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(<scope>!=nil?<scope>.text:nil)"
lexerRuleLabelPropertyRef_int(scope,attr) ::="(<scope>!=nil?[<scope>.text integerValue]:0)"

// Somebody may ref $template or $tree or $stop within
 a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType> *)retval.start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType> *)retval.stopToken)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType> *)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
[[input getTokenStream] toStringFromStart:[[input getTreeAdaptor]
getTokenStartIndex:retval.start.token.startIndex]
                                  ToEnd:[[input getTreeAdaptor] getTokenStopIndex:retval.start.token.stopIndex]]
<else>
[input toStringFromToken:retval.start ToToken:[input LT:-1]]
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

/* hideous: find a way to cut down on the number of templates to support read/write access */
/* TODO: also, which ones are valid to write to? ask Ter */
lexerRuleSetPropertyRef_text(scope,attr,expr) ::= "state.text = <expr>;"
lexerRuleSetPropertyRef_type(scope,attr,expr) ::= "_type"
lexerRuleSetPropertyRef_line(scope,attr,expr) ::= "state.tokenStartLine"
lexerRuleSetPropertyRef_pos(scope,attr,expr)
```

```
 ::= "state.tokenStartCharPositionInLine"
lexerRuleSetPropertyRef_index(scope,attr,expr) ::= "-1" /* undefined token index in lexer */
lexerRuleSetPropertyRef_channel(scope,attr,expr) ::= "state.channel=<expr>;"
lexerRuleSetPropertyRef_start(scope,attr,expr) ::= "state.tokenStartCharIndex"
lexerRuleSetPropertyRef_stop(scope,attr,expr) ::= "(input.index-1)"


lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "state.type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(input.index-1)"
lexerRulePropertyRef_int(scope,attr) ::= "[<scope>.text integerValue]"

//
 setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.start =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;" /* "<\n>#error StringTemplates are
unsupported<\n>" */


/** How to execute an action */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
   <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
static ANTLRBitSet *<name>;
static const unsigned long long <name>_data[] = { <words64:{it | <it>LL};separator=", ">};<\n>
>>

bitsetInit(name, words64) ::= <<
<name> = [[ANTLRBitSet newBitSetWithBits:(const unsigned long long *)<name>_data
Count:(NSUInteger)<length(words64)>] retain];<\n>
>>

codeFileExtension() ::= ".m"

true_value() ::= "YES"
false_value() ::= "NO"
```

Found in path(s):
*
 /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ObjC.stg
No license file was found, but licenses were detected in source scan.

```
/** How to generate rules derived from left-recursive rules.
*  These rely on recRuleDefArg(), recRuleAltPredicate(),
*  recRuleArg(), recRuleSetResultAction(), recRuleSetReturnAction()
*  templates in main language.stg
*/
group LeftRecursiveRules;
```

```
recRuleName(ruleName) ::= "<ruleName>_"
recPrimaryName(ruleName) ::= "<ruleName>_primary"

recRuleStart(ruleName, minPrec, userRetvals, userRetvalAssignments) ::= <<
<ruleName><if(userRetvals)> returns [<userRetvals>]<endif>
 : <recRuleName(...)>[<minPrec>]
    <if(userRetvals)>
   {
       <userRetvalAssignments; separator="\n">
   }
   <endif>
 ;
>>

recRule(ruleName, precArgDef, argName, alts, setResultAction, buildAST,
      userRetvals, userRetvalAssignments) ::= <<
<recRuleName(...)>[<precArgDef>]<if(userRetvals)> returns [<userRetvals>]<endif>
   :   <recPrimaryName(...)>
       <if(buildAST)>
       {
          <setResultAction>
       }
       <endif>
       <if(userRetvals)>
       {
          <userRetvalAssignments; separator="\n">
       }
       <endif>
       ( options {backtrack=false;}
       : ( options {backtrack=false;}
         : <alts; separator="\n         | ">
         )
       )*
   ;
>>

recPrimaryRule(ruleName, alts, userRetvals) ::= <<
<recPrimaryName(...)><if(userRetvals)> returns [<userRetvals>]<endif>
options {backtrack=true;}
   : <alts; separator="\n    | ">
   ;
>>

recRuleAlt(alt, pred) ::= "{<pred>}?=> <alt>"

recRuleRef(ruleName, arg) ::= "<recRuleName(...)>[<arg>]"
```

Found
 in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/LeftRecursiveRules.stg
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to normal Objective-C output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
@headerFile.imports() ::= <<
<@super.imports()>
#import \<ANTLR/Debug.h>
>>

@parserHeaderFile.memVars() ::= <<
NSInteger ruleLevel;
NSArray *ruleNames;
>>

@parserHeaderFile.methodsDecl() ::= <<
```

```
-(BOOL) evalPredicate:(NSString *)predicate matched:(BOOL)result;<\n>
>>

@genericParser.methods() ::= <<
<if(grammar.grammarIsRoot)>
AMutableArray
 *ruleNames = [AMutableArray arrayWithArray:{
   @"invalidRule", <grammar.allImportedRules:{rST | @"<rST.name>"}; wrap=@"\n    ", separator=", ">
};<\n>
<endif>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
   ruleLevel = 0;
- (NSInteger) getRuleLevel { return ruleLevel; }
- (void) incRuleLevel { ruleLevel++; }
- (void) decRuleLevel { ruleLevel--; }
<if(profile)>
   <ctorForProfilingRootGrammar()>
<else>
   <ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else> <! imported grammar !>
- (NSInteger) getRuleLevel
{
   return <grammar.delegators:{g| <g:delegateName()>}>.getRuleLevel();
}<\n>

- (void) incRuleLevel
{
   <grammar.delegators:{g| <g:delegateName()>}>.incRuleLevel();
}<\n>
- (void) decRuleLevel
{
   <grammar.delegators:{g| <g:delegateName()>}>.decRuleLevel();
}<\n>
   <ctorForDelegateGrammar()>
<endif>
<if(profile)>
- (BOOL) alreadyParsedRule:(id<IntStream>) input Index:(NSInteger) ruleIndex
{
   [(Profiler)dbg examineRuleMemoization:input,
 ruleIndex, <grammar.composite.rootGrammar.recognizerName>.ruleNames objectAtIndex:ruleIndex];
   return super.alreadyParsedRule(input, ruleIndex);
}<\n>
- (void) memoize:(id<IntStream>)input RuleIndex:(NSInteger)ruleIndex StartIndex:(NSInteger)ruleStartIndex
{
   [((Profiler)dbg) memoize:input RuleIndex:ruleIndex StartIndex:ruleStartIndex
[<grammar.composite.rootGrammar.recognizerName> ruleNames[ruleIndex]];
```

```
  [super memoize:input RuleIndex:ruleIndex StartIndex:ruleStartIndex];
}<\n>
<endif>
- (BOOL) evalPredicate:(BOOL)result Pred:(NSString *)predicate
{
  [dbg semanticPredicate:result Pred:predicate];
  return result;
}<\n>
>>

@genericParser.init() ::= <<
ruleNames = [NSArray arrayWithObjects:<rules:{rST | @"<rST.ruleName>"}; separator=", ", wrap="\n ">,
nil];<\n>
>>

@genericParser.dealloc() ::= <<
[ruleNames release];<\n>
>>

@genericParser.methods() ::= <<
-(BOOL) evalPredicate:(NSString *)predicate matched:(BOOL)result
{
 [debugListener semanticPredicate:predicate
 matched:result];
 return result;
}<\n>
>>

/* bug: can't use @super.superClassName()> */
@parserHeaderFile.superClassName() ::= "Debug<if(TREE_PARSER)>Tree<endif>Parser"

@rule.preamble() ::= <<
@try { [debugListener enterRule:@"<ruleName>"];
if ( ruleLevel==0 ) [debugListener commence];
ruleLevel++;
[debugListener locationLine:<ruleDescriptor.tree.line> column:<ruleDescriptor.tree.column>];<\n>
>>

@rule.postamble() ::= <<
[debugListener locationLine:<ruleDescriptor.EORNode.line> column:<ruleDescriptor.EORNode.column>];<\n>
}
@finally {
  [debugListener exitRule:@"<ruleName>"];
  ruleLevel--;
  if ( ruleLevel==0 ) [debugListener terminate];
}<\n>
>>
```

```
/* these are handled in the runtime for now.
* stinks, but that's the easiest way to avoid having to generate two
* methods for each synpred

@synpred.start() ::= "[debugListener beginBacktrack:state.backtracking];"

@synpred.stop() ::= "[debugListener endBacktrack:state.backtracking wasSuccessful:success];"

*/

// Common debug
 event triggers used by region overrides below

enterSubRule() ::=
   "@try { [debugListener enterSubRule:<decisionNumber>];<\n>"

exitSubRule() ::=
   "} @finally { [debugListener exitSubRule:<decisionNumber>]; }<\n>"

enterDecision() ::=
   "@try { [debugListener enterDecision:<decisionNumber>];<\n>"

exitDecision() ::=
   "} @finally { [debugListener exitDecision:<decisionNumber>]; }<\n>"

enterAlt(n) ::= "[debugListener enterAlt:<n>];<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision()
```

```
 ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
   "[debugListener recognitionException:eee];<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
   "[debugListener locationLine:<it.line> column:<it.pos>];"

@matchSet.mismatchedSetException() ::=
   "[debugListener recognitionException:mse];"

@dfaState.noViableAltException() ::= "[debugListener recognitionException:nvae];"

@dfaStateSwitch.noViableAltException() ::= "[debugListener recognitionException:nvae];"

dfaDecision(decisionNumber,description) ::= <<
@try {
   // isCyclicDecision is only necessary for the Profiler. Which I didn't do, yet.
   // isCyclicDecision = YES;
   <super.dfaDecision(...)>
}
@catch
 (NoViableAltException *nvae) {
   [debugListener recognitionException:nvae];
   @throw nvae;
}
>>

@cyclicDFA.errorMethod() ::= <<
-(void) error:(NoViableAltException *)nvae
{
   [[recognizer debugListener] recognitionException:nvae];
}
>>

/** Force predicate validation to trigger an event */
```

```
evalPredicate(pred,description) ::= <<
[self evalPredicate:@"<description>" result:<pred>];
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/Dbg.stg
No license file was found, but licenses were detected in source scan.

```
scalaTypeInitMap ::= [
 "Int":"0",
 "Long":"0",
 "Float":"0.0f",
 "Double":"0.0",
 "Boolean":"false",
 "Byte":"0",
 "Short":"0",
 "Char":"0",
 default:"null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,

   bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>

<@imports>
import org.antlr.runtime._
<if(TREE_PARSER)>
import org.antlr.runtime.tree._
<endif>
<@end>

<docComment>
<recognizer>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
     superClass="Lexer") ::= <<
object <grammar.recognizerName> {
   <tokens:{it | val <it.name> = <it.type>}; separator="\n">

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
}

class <grammar.recognizerName>(input: CharStream, state<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>: RecognizerSharedState) extends <@superClassName><superClass><@end>(input,
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>) {
   import <grammar.recognizerName>._
    <actions.lexer.members>

   // delegates
   <grammar.delegates:
        {g|<g.recognizerName> <g:delegateName()>}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|<g.recognizerName> <g:delegateName()>}; separator="\n">
   <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

   <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScope()><endif>}>

   def this(input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>: CharStream) =
       this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>)

<if(memoize)>
<if(grammar.grammarIsRoot)>
       state.ruleMemo = new Array[java.util.Map[_,_]](<numRules>+1)<\n> <! index from 1..n !>
<endif>
<endif>
       <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this)}; separator="\n">
       <grammar.delegators:
        {g|this.<g:delegateName()>
 = <g:delegateName()>}; separator="\n">
       <last(grammar.delegators):{g|gParent = <g:delegateName()>}>

   override def getGrammarFileName = "<fileName>"

<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>
   <cyclicDFAs:{dfa | private val dfa<dfa.decisionNumber> = new
<grammar.recognizerName>.DFA<dfa.decisionNumber>(this)}; separator="\n">
}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
override def nextToken(): Token = {
   while (true) {
       if ( input.LA(1)==CharStream.EOF ) {
           var eof: Token = new CommonToken((CharStream)input,Token.EOF,

        Token.DEFAULT_CHANNEL,
                                  input.index(),input.index())
```

```
          eof.setLine(getLine())
          eof.setCharPositionInLine(getCharPositionInLine())
          return eof
        }
        state.token = null
 state.channel = Token.DEFAULT_CHANNEL
        state.tokenStartCharIndex = input.index()
        state.tokenStartCharPositionInLine = input.getCharPositionInLine()
        state.tokenStartLine = input.getLine()
 state.text = null
        try {
          val m = input.mark()
          state.backtracking=1 <! means we won't throw slow exception !>
          state.failed=false
          mTokens()
          state.backtracking=0
          <! mTokens backtracks with synpred at backtracking==2
             and we set the synpredgate to allow actions at level 1. !>
          if ( state.failed ) {
              input.rewind(m)
              input.consume() <! advance one char and try again !>
          }
          else
 {
              emit()
              return state.token
          }
        }
        catch {
          case re: RecognitionException =>
          // shouldn't happen in backtracking mode, but...
          reportError(re)
          recover(re)
        }
    }
}

override def memoize(input: IntStream,
  ruleIndex: Int,
  ruleStartIndex: Int) = {
if ( state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex)
}

override def alreadyParsedRule(input: IntStream, ruleIndex: Int):Boolean {
if ( state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex)
return false
}
>>
```

```
actionGate() ::= "state.backtracking==0"

filteringActionGate() ::= "state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
            labelType, members, rewriteElementType,
            filterMode, ASTLabelType="Object") ::= <<
object <grammar.recognizerName>
 {
<if(grammar.grammarIsRoot)>
   val tokenNames = Array(
      "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
   )<\n>
<endif>

   <tokens:{it | val <it.name> = <it.type>}; separator="\n">

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                 words64=it.bits)>}>
}

class <grammar.recognizerName>(input: <inputStreamType>, state<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>: RecognizerSharedState) extends <@superClassName><superClass><@end>(input, state) {
   import <grammar.recognizerName>._
   // delegates
   <grammar.delegates:
        {g|public <g.recognizerName> <g:delegateName()>}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|public <g.recognizerName> <g:delegateName()>}; separator="\n">
   <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

   <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScope()><endif>}>

    <@members>
   <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
   def this(input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>: <inputStreamType>) =
      this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>)

      <parserCtorBody()>
      <grammar.directDelegates:
       {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this)}; separator="\n">
```

```
    <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>};
separator="\n">
    <last(grammar.delegators):{g|gParent = <g:delegateName()>}>
  <@end>

  override def getTokenNames: Array[String] = tokenNames
  override def getGrammarFileName = "<fileName>"

  <members>

  <rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
   appear to be defined in this recognizer. !>
   // Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
  @throws(classOf[RecognitionException])
  def <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>): <returnType()> = \{
<if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>}; separator=", ">) \}}; separator="\n">

  <synpreds:{p | <synpred(p)>}>

  <cyclicDFAs:{dfa | private val dfa<dfa.decisionNumber> = new
<grammar.recognizerName>.DFA<dfa.decisionNumber>(this)}; separator="\n">
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new Array[java.util.Map[_,_]](<length(grammar.allImportedRules)>+1)<\n> <! index from
1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
     ASTLabelType="Object", superClass="Parser",
 labelType="Token",
     members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="TokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
```

```
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
          numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
 members={<actions.treeparser.members>}
          ) ::= <<
<genericParser(inputStreamType="TreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
 *  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
@throws(classOf[RecognitionException])
def <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>): Unit = {
   <ruleLabelDefs()>
<if(trace)>
   traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
   try {
       <block>
   }
   finally {
       traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
   }
<else>
   <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
final def <name>(): Boolean = {
   state.backtracking+=1
   <@start()>
   val start = input.mark()
   try {
       <name>_fragment() // can never throw exception
   } catch {
       case re: RecognitionException =>
       System.err.println("impossible: "+re)
   }
   val success = !state.failed
   input.rewind(start)
   <@stop()>
   state.backtracking-=1
   state.failed=false
    success
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) return <ruleReturnValue()><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true; return <ruleReturnValue()>}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
@throws(classOf[RecognitionException])
final
 def <ruleName>(<ruleDescriptor.parameterScope:parameterScope()>): <returnType()> = {
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>)<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   try {
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
   }
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   catch {
       case re: RecognitionException =>
       reportError(re)
       recover(input,re)
 <@setErrorReturnValue()>
   }<\n>
<endif>
<endif>
<endif>
   finally {
       <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
       <memoize()>
       <ruleScopeCleanUp()>
       <finally>
   }
   <@postamble()>

 return <ruleReturnValue()>
}
// $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
   <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
val retval = new <returnType()>()
retval.start = input.LT(1)<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
var <a.name>: <a.type> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>
```

```
}>
<endif>
<if(memoize)>
val <ruleDescriptor.name>_StartIndex = input.index()
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it | <it>_stack.push(new <it>_scope())}; separator="\n">
<ruleDescriptor.ruleScope:{it | <it.name>_stack.push(new <it.name>_scope())}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it | <it>_stack.pop()}; separator="\n">
<ruleDescriptor.ruleScope:{it | <it.name>_stack.pop()}; separator="\n">
>>


ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
    :{it | var <it.label.text>: <labelType> = null}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it | var list_<it.label.text>: java.util.List=null}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|var <ll.label.text>: RuleReturnScope = null}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it | var <it.label.text>: <labelType>=null}; separator="\n"
>
<ruleDescriptor.charLabels:{it | int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{it | var list_<it.label.text>: java.util.List=null}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
```

```
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp()
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1)<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex) }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
@throws(classOf[RecognitionException])
final def m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>): Unit = {
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>)<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   try {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       try <block><\n>
<else>
       var _type = <ruleName>

  var _channel = BaseRecognizer.DEFAULT_TOKEN_CHANNEL
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       try <block>
```

```
        <ruleCleanUp()>
        state.`type` = _type
        state.channel = _channel
        <(ruleDescriptor.actions.after):execAction()>
<endif>
    }
    finally {
        <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>)<endif>
        <ruleScopeCleanUp()>
        <memoize()>
    }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
@throws(classOf[RecognitionException])
def mTokens(): Unit = {
    <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber>
 = <maxAlt>
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
alt<decisionNumber> match {
    <alts:{a | <altSwitchCase(i,a)>}>
    case _ =>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber> = <maxAlt>
```

```
<decls>
<@predecision()>
<decision>
<@postdecision()>
alt<decisionNumber> match {
   <alts:{a | <altSwitchCase(i,a)>}>
   case _ =>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives
 */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var cnt<decisionNumber>: Int = 0
<decls>
<@preloop()>
var loop<decisionNumber>_quitflag = false
while (!loop<decisionNumber>_quitflag) {
   var alt<decisionNumber>:Int = <maxAlt>
   <@predecision()>
   <decision>
   <@postdecision()>
   alt<decisionNumber> match {
       <alts:{a | <altSwitchCase(i,a)>}>
 case _ =>
     if ( cnt<decisionNumber> >= 1 ) loop<decisionNumber>_quitflag = true
     else {
     <ruleBacktrackFailure()>
           val eee = new EarlyExitException(<decisionNumber>, input)
```

```
          <@earlyExitException()>
          throw eee
     }
   }
   cnt<decisionNumber>+=1
}
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
//
 <fileName>:<description>
<decls>
<@preloop()>
var loop<decisionNumber>_quitflag = false
while (!loop<decisionNumber>_quitflag) {
   var alt<decisionNumber>:Int = <maxAlt>
   <@predecision()>
   <decision>
   <@postdecision()>
   alt<decisionNumber> match {
       <alts:{a | <altSwitchCase(i,a)>}>
 case _ => loop<decisionNumber>_quitflag = true
   }
}
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum, alt) ::= <<
case <altNum> =>
   <@prealt()>
   <alt>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew)
 ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label>=<endif>`match`(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex>)<if(label)>
.asInstanceOf[<labelType>]<endif>
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new java.util.ArrayList()
list_<label>.add(<elem>)<\n>
>>

/** match a character */
charRef(char,label) ::= <<
```

```
<if(label)>
<label> = input.LA(1)<\n>
<endif>
`match`(<char>)
<checkRuleBacktrackFailure()>
>>

/**
 match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1)<\n>
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1)<\n>
<else>
<label>=input.LT(1).asInstanceOf[<labelType>]<\n>
<endif>
<endif>
if ( <s> ) {
  input.consume()
  <postmatchCode>
<if(!LEXER)>
  state.errorRecovery=false<\n>
<endif>
  <if(backtracking)>state.failed=false<endif>
}
else {
  <ruleBacktrackFailure()>
  val mse = new MismatchedSetException(null,input)
  <@mismatchedSetException()>
<if(LEXER)>
  recover(mse)
  throw mse
<else>
  throw mse
  <! use following code to make it recover inline; remove throw mse;
  recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>)
  !>
<endif>
}<\n>
>>
```

```
matchRuleBlockSet
 ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
val <label>Start = getCharIndex()
`match`(<string>)
<checkRuleBacktrackFailure()>
val <label>StartLine<elementIndex> = getLine()
val <label>StartCharPos<elementIndex> = getCharPositionInLine()
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL, <label>Start,
getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
`match`(<string>)
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=input.LT(1).asInstanceOf[<labelType>]<\n>
<endif>
matchAny(input)
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/**
 Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1)<\n>
<endif>
matchAny()
<checkRuleBacktrackFailure()>
```

```
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">)<\n>
state._fsp-=1
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*
  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
val <label>Start<elementIndex> = getCharIndex()
val <label>StartLine<elementIndex> = getLine()
val <label>StartCharPos<elementIndex> = getCharPositionInLine()
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<checkRuleBacktrackFailure()>
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<checkRuleBacktrackFailure()>
<endif>
```

```
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/**
 EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
val <label>Start<elementIndex> = getCharIndex()
val <label>StartLine<elementIndex> = getLine()
val <label>StartCharPos<elementIndex> = getCharPositionInLine()
`match`(EOF)
<checkRuleBacktrackFailure()>
val <label> = new <labelType>(input, EOF, Token.DEFAULT_CHANNEL, <label>Start<elementIndex>,
getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
`match`(EOF)
<checkRuleBacktrackFailure()>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()                       ::= "int <recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root,
 actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   `match`(input, Token.DOWN, null)
   <checkRuleBacktrackFailure()>
   <children:element()>
   `match`(input, Token.UP, null)
   <checkRuleBacktrackFailure()>
}
```

```
<else>
`match`(input, Token.DOWN, null)
<checkRuleBacktrackFailure()>
<children:element()>
`match`(input, Token.UP, null)
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
   <ruleBacktrackFailure()>
   throw new FailedPredicateException(input, "<ruleName>", "<description>")
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber> = input.LA(<k>)<\n>
<edges;
 separator="\nelse ">
else {
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   val nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input)<\n>
   <@noViableAltException()>
   throw nvae<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber> = input.LA(<k>)<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
```

```
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer
 so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
val LA<decisionNumber>_<stateNumber> = input.LA(<k>)<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
else {
   alt<decisionNumber>=<eotPredictsAlt>
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The
 code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
input.LA(<k>) match {
<edges; separator="\n">
case _ =>
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   val nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input)<\n>
   <@noViableAltException()>
   throw nvae<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
input.LA(<k>) match {
   <edges; separator="\n">
   case _ =>
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
input.LA(<k>) match {
<edges; separator="\n"><\n>
case _ =>
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
case <labels:{it | <it>}; separator=" | "> =>
   {
   <targetState>
    }
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input)
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
```

```
cyclicDFA(dfa) ::= <<
val DFA<dfa.decisionNumber>_eotS =
  "<dfa.javaCompressedEOT; wrap="\"+\n    \"">"
val DFA<dfa.decisionNumber>_eofS =
  "<dfa.javaCompressedEOF; wrap="\"+\n    \"">"
val DFA<dfa.decisionNumber>_minS =
  "<dfa.javaCompressedMin;
 wrap="\"+\n    \"">"
val DFA<dfa.decisionNumber>_maxS =
  "<dfa.javaCompressedMax; wrap="\"+\n    \"">"
val DFA<dfa.decisionNumber>_acceptS =
  "<dfa.javaCompressedAccept; wrap="\"+\n    \"">"
val DFA<dfa.decisionNumber>_specialS =
  "<dfa.javaCompressedSpecial; wrap="\"+\n    \"">}>"
val DFA<dfa.decisionNumber>_transitionS: Array[String] = Array(
    <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
)

val DFA<dfa.decisionNumber>_eot: Array[Short] = DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eotS)
val DFA<dfa.decisionNumber>_eof: Array[Short] = DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eofS)
val DFA<dfa.decisionNumber>_min: Array[Char] =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS)
val DFA<dfa.decisionNumber>_max: Array[Char] =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS)
val DFA<dfa.decisionNumber>_accept: Array[Short] =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_acceptS)
val
 DFA<dfa.decisionNumber>_special: Array[Short] =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_specialS)
val DFA<dfa.decisionNumber>_transition = new
Array[Array[Short]](DFA<dfa.decisionNumber>_transitionS.length)

for (i \<- DFA<dfa.decisionNumber>_transition.indices) {
  DFA<dfa.decisionNumber>_transition(i) =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_transitionS(i))
}

class DFA<dfa.decisionNumber> extends DFA {

  def this(recognizer: BaseRecognizer) = {
    this()
    this.recognizer = recognizer
    this.decisionNumber = <dfa.decisionNumber>
    this.eot = DFA<dfa.decisionNumber>_eot
    this.eof = DFA<dfa.decisionNumber>_eof
    this.min = DFA<dfa.decisionNumber>_min
    this.max = DFA<dfa.decisionNumber>_max
    this.accept = DFA<dfa.decisionNumber>_accept
```

```
        this.special = DFA<dfa.decisionNumber>_special
        this.transition = DFA<dfa.decisionNumber>_transition
    }
    override def getDescription = "<dfa.description>"
     <@errorMethod()>
<if(dfa.specialStateSTs)>
    @throws(classOf[NoViableAltException])
    override def specialStateTransition(s: Int, _input: IntStream):Int = {
        <if(LEXER)>
        val input = _input
        <endif>
        <if(PARSER)>
        val input = _input.asInstanceOf[TokenStream]
        <endif>
        <if(TREE_PARSER)>
        val input = _input.asInstanceOf[TreeNodeStream]
        <endif>
     val _s = s
        s match {
        <dfa.specialStateSTs:{state |
        case <i0> => <! compressed special state numbers 0..n-1 !>
            <state>}; separator="\n">
        case _ =>
        }
<if(backtracking)>
        if (state.backtracking>0) {state.failed=true; return -1}<\n>
<endif>
        val nvae = new NoViableAltException(getDescription(), <dfa.decisionNumber>, _s, input)
        error(nvae)
        throw nvae
    }<\n>
<endif>
}<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState)
 ::= <<
val LA<decisionNumber>_<stateNumber>: Int = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
val index<decisionNumber>_<stateNumber>: Int = input.index()
input.rewind()<\n>
<endif>
s = -1
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
```

```
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if ( s>=0 ) return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber><\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right)
 ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(pred,\"\")>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber> >= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>) >=<lower>
&& input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= "<ranges; separator=\"||\">"
```

```
// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
class <scope.name>_scope
 {
   <scope.attributes:{it | var <it.name>: <it.type> = _}; separator="\n">
}
val <scope.name>_stack = new collection.mutable.Stack[<scope.name>_scope]<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
class <scope.name>_scope {
   <scope.attributes:{it | var <it.name>: <it.type> = _}; separator="\n">
}
val <scope.name>_stack = new collection.mutable.Stack[<scope.name>_scope]<\n>
<endif>
>>

returnStructName(r) ::= "<r.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
Unit
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
Unit
<endif>
```

```
<endif>
>>

delegateName(d)
 ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<scalaTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
var <label.label.text>: <ruleLabelType(referencedRule=label.referencedRule)> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))><\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*  TODO(matthewlloyd): make this static
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
final class <ruleDescriptor:returnStructName()> extends
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope {
   <scope.attributes:{it | var <it.name>: <it.type> = _}; separator="\n">
   <@ruleReturnMembers()>
}
<endif>
>>

parameterScope(scope)
 ::= <<
<scope.attributes:{it | <it.name>: <it.type>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>"

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack(<scope>_stack.size-<negIndex>-1).<attr.name>
<else>
```

```
<if(index)>
<scope>_stack(<index>).<attr.name>
<else>
<scope>_stack.top.<attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack(<scope>_stack.size-<negIndex>-1).<attr.name> = <expr>
<else>
<if(index)>
<scope>_stack(<index>).<attr.name> = <expr>
<else>
<scope>_stack.top.<attr.name> = <expr>
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/**
 reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(if (<scope>!=null) <scope>.<attr.name> else <initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>
<else>
```

```
<attr.name> =<expr>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(if (<scope>!=null) <scope>.getText() else null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(if (<scope>!=null) <scope>.getType() else 0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(if (<scope>!=null)
 <scope>.getLine() else 0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(if (<scope>!=null) <scope>.getCharPositionInLine() else 0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(if (<scope>!=null) <scope>.getChannel() else 0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(if (<scope>!=null) <scope>.getTokenIndex() else 0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(if (<scope>!=null) Integer.valueOf(<scope>.getText()) else 0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(if (<scope>!=null) <scope>.start.asInstanceOf[<labelType>] else null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(if (<scope>!=null) <scope>.stop.asInstanceOf[<labelType>] else null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(if (<scope>!=null) <scope>.tree.asInstanceOf[<ASTLabelType>] else
null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(if (<scope>!=null) (input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
  input.getTreeAdaptor().getTokenStopIndex(<scope>.start))) else null)
<else>
(if (<scope>!=null) input.toString(<scope>.start,<scope>.stop) else null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "(if (<scope>!=null) <scope>.st else null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(if (<scope>!=null) <scope>.getType() else 0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(if (<scope>!=null) <scope>.getLine() else 0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
   "(if (<scope>!=null) <scope>.getCharPositionInLine() else -1)"
```

```
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(if (<scope>!=null) <scope>.getChannel() else 0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::=
   "(if (<scope>!=null) <scope>.getTokenIndex() else 0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
   "(if (<scope>!=null) <scope>.getText() else null)"
lexerRuleLabelPropertyRef_int(scope,attr)
 ::=
   "(if (<scope>!=null) Integer.valueOf(<scope>.getText()) else 0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "(retval.start.asInstanceOf[<labelType>])"
rulePropertyRef_stop(scope,attr) ::= "(retval.stop.asInstanceOf[<labelType>])"
rulePropertyRef_tree(scope,attr) ::= "(retval.tree.asInstanceOf[<ASTLabelType>])"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(retval.start),
 input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr)
 ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "Integer.valueOf(<scope>.getText())"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
```

```
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
val <name> = new BitSet(Array[Long](<words64:{it | <it>L};separator=",">))<\n>
>>

codeFileExtension()
 ::= ".scala"

true_value() ::= "true"
false_value() ::= "false"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Scala/Scala.stg
No license file was found, but licenses were detected in source scan.

```
/*
New style messages. This file contains the actual layout of the messages emitted by ANTLR.
The text itself is coming out of the languages/*stg files, according to the chosen locale.
This file contains the default format ANTLR uses.
*/

group antlr;

location(file, line, column) ::= "<file>(<line>,<column>)"

message(id, text) ::= "error <id> : <text>"

report(location, message, type) ::= "<location> : <type> <message.id> : <message.text>"

wantsSingleLineMessage() ::= "true"
```

Found
 in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/messages/formats/vs2005.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
 *  is combined into tokenRef, ruleRef, ...
*/

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<if(!ruleDescriptor.isSynPred)>
<ASTLabelType> _first_0 = default(<ASTLabelType>);
<ASTLabelType> _last = default(<ASTLabelType>);
<endif>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel=false, treeLevel=false) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(rewriteMode)>
retval.Tree = (<ASTLabelType>)_first_0;
if (adaptor.GetParent(retval.Tree)!=null && adaptor.IsNil(adaptor.GetParent(retval.Tree)))
   retval.Tree = (<ASTLabelType>)adaptor.GetParent(retval.Tree);
<endif>
<if(backtracking)>}<endif>
<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
```

```
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel)
 ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = default(<ASTLabelType>);
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(root.el.rule)>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>.Tree;
<else>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down) {
   Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<enclosingTreeLevel>,
 root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}
<else>
<super.tree(...)>
<endif>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
```

```
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<else>
<super.tokenRefBang(...)>
<endif>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text;
 format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
<endif>
<else>
<super.tokenRef(...)>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRef(...)>
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefAndListLabel(...)>
<endif>
>>

/** ^(ID ...) auto construct */
```

```
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree
 = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
<else>
<super.tokenRefRuleRoot(...)>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefRuleRootAndListLabel(...)>
<endif>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if
 (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.DupTree(<label>);
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
```

```
<endif>
<else>
<super.wildcard(...)>
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text;
 format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSet(...)>
<endif>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<if(!ruleDescriptor.isSynPred)>
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
<else>
<super.matchRuleBlockSet(...)>
<endif>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
<else>
```

```
<super.matchSetBang(...)>
<endif>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(!ruleDescriptor.isSynPred)>
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text;
 format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSetRuleRoot(...)>
<endif>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<treeLevel>, <label>.Tree);
<else> <! rewrite mode !>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>.Tree;
<endif>
<else>
<super.ruleRef(...)>
<endif>
>>

/** x+=rule auto construct
```

```
 */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefAndListLabel(...)>
<endif>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_<treeLevel>);
<endif>
<else>
<super.ruleRefRuleRoot(...)>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefRuleRootAndListLabel(...)>
<endif>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last
 = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
<else>
<super.ruleRefTrack(...)>
<endif>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
```

```
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
<else>
<super.ruleRefTrackAndListLabel(...)>
<endif>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
<else>
<super.ruleRefRuleRootTrack(...)>
<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
<else>
<super.ruleRefRuleRootTrackAndListLabel(...)>
<endif>
>>

/** Streams for token refs are tree nodes now;
 override to
*  change NextToken to NextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,args) ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif>stream_<token>.NextNode())
<else>
stream_<token>.NextNode()
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!ruleDescriptor.isSynPred)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
```

```
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

```
/*
[The "BSD license"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template overrides to add debugging to normal C output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
const char *
ruleNames[] =
 {
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n  ", separator=", ">
 };<\n>
<endif>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
static ANTLR3_UINT32
 ruleLevel = 0;
static ANTLR3_UINT32 getRuleLevel()
{
 return ruleLevel;
}
static void incRuleLevel()
{
 ruleLevel++;
}
static void decRuleLevel()
{
 ruleLevel--;
}
<else> <! imported grammar !>
static ANTLR3_UINT32
getRuleLevel()
{
 return <grammar.delegators:{g| <g:delegateName()>}>->getRuleLevel();
}
static void incRuleLevel()
{
 <grammar.delegators:{g| <g:delegateName()>}>->incRuleLevel();
}
static void
decRuleLevel()
{
 <grammar.delegators:{g| <g:delegateName()>}>.decRuleLevel();
}
<endif>
<if(profile)>
// Profiling not yet implemented for C target
//
<endif>
<if(grammar.grammarIsRoot)>
<ctorForPredefinedListener()>
<else>
<ctorForDelegateGrammar()>
<endif>

static ANTLR3_BOOLEAN
evalPredicate(p<name> ctx, ANTLR3_BOOLEAN result, const char * predicate)
{
   DBG->semanticPredicate(DBG, result, predicate);
   return result;
```

```
}<\n>
>>

@genericParser.debugStuff() ::= <<
<if(grammar.grammarIsRoot)>
<createListenerAndHandshake()>
<endif>
>>

ctorForProfilingRootGrammar() ::=
 <<
>>

/** Basically we don't want to set any dbg listeners as root will have it. */
ctorForDelegateGrammar() ::= <<

>>

ctorForPredefinedListener() ::= <<

>>

createListenerAndHandshake() ::= <<
{
 // DEBUG MODE code
 //
 pANTLR3_DEBUG_EVENT_LISTENER  proxy;
 proxy = antlr3DebugListenerNew();
 proxy->grammarFileName = INPUT->tokenSource->strFactory->newStr8(INPUT->tokenSource->strFactory,
(pANTLR3_UINT8)ctx->getGrammarFileName());

<if(TREE_PARSER)>
 proxy->adaptor = ADAPTOR;
<endif>
 PARSER->setDebugListener(PARSER, proxy);

 // Try to connect to the debugger (waits forever for a connection)
 //
 proxy->handshake(proxy);

 // End DEBUG MODE code
 //
}
>>



@rule.preamble() ::= <<
if ( getRuleLevel()==0 )
```

```
{
 DBG->commence(DBG);
}
DBG->enterRule(DBG, getGrammarFileName(), (const char *)"<ruleName>");
incRuleLevel();
DBG->location(DBG, <ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>);<\n>
>>

@rule.postamble() ::= <<
DBG->location(DBG, <ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>);<\n>
DBG->exitRule(DBG,
 getGrammarFileName(), (const char *)"<ruleName>");
decRuleLevel();
if ( getRuleLevel()==0 )
{
 DBG->terminate(DBG);
}
<\n>
>>

@checkRuleBacktrackFailure.debugClean() ::= <<
DBG->exitRule(DBG, getGrammarFileName(), (const char *)"<ruleName>");
decRuleLevel();
>>

@synpred.start() ::= "DBG->beginBacktrack(DBG, BACKTRACKING);"

@synpred.stop() ::= "DBG->endBacktrack(DBG, BACKTRACKING, success);"

// Common debug event triggers used by region overrides below

enterSubRule() ::=
   "DBG->enterSubRule(DBG, <decisionNumber>);<\n>"

exitSubRule() ::=
   "DBG->exitSubRule(DBG, <decisionNumber>);<\n>"

enterDecision() ::=
   "DBG->enterDecision(DBG, <decisionNumber>);<\n>"

exitDecision() ::=
   "DBG->exitDecision(DBG, <decisionNumber>);<\n>"

enterAlt(n) ::= "DBG->enterAlt(DBG, <n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"
```

```
@block.postdecision()
 ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
   "DBG->recognitionException(DBG, EXCEPTION);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(altNum)>"

@element.prematch() ::=
   "DBG->location(DBG, <e.line>, <e.pos>);" // e is arg of element

@matchSet.mismatchedSetException() ::=
    "DBG->recognitionException(DBG, EXCEPTION);"

@newNVException.noViableAltException() ::= "DBG->recognitionException(DBG, EXCEPTION);"

dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(ctx, RECOGNIZER, ISTREAM,
&cdfa<decisionNumber>);
if  (HASEXCEPTION())
{
 DBG->recognitionException(DBG, EXCEPTION);
   goto rule<ruleDescriptor.name>Ex;
}
<checkRuleBacktrackFailure()>
>>

@cyclicDFA.errorMethod() ::= <<
//static void
//dfaError(p<name> ctx)
//{
//    DBG->recognitionException(DBG, EXCEPTION);
//}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
evalPredicate(ctx, <pred>, (const char *)"<description>")
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/Dbg.stg
No license file was found, but licenses were detected in source scan.

```
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
@genericParserHeaderFile.memVars() ::= <<
/* AST parserHeaderFile.memVars */
NSInteger ruleLevel;
NSArray *ruleNames;
<@super.memVars()>  /* AST super.memVars */
<parserMemVars()>   /* AST parserMemVars */
>>

@genericParserHeaderFile.properties() ::= <<
/* AST parserHeaderFile.properties */
<@super.properties()>  /* AST super.properties */
<parserProperties()>   /* AST parserproperties */
>>

@genericParserHeaderFile.methodsDecl() ::= <<
/* AST parserHeaderFile.methodsDecl
 */
<@super.methodsDecl()>  /* AST super.methodsDecl */
<parserMethodsDecl()>   /* AST parsermethodsDecl */
>>

@genericParser.synthesize() ::= <<
/* AST genericParser.synthesize */
<@super.synthesize()>
<parserSynthesize()>
>>

@genericParser.methods() ::= <<
/* AST genericParser.methods */
<@super.methods()>
<parserMethods()>
>>

/* additional init code for tree support */
@genericParser.init() ::= <<
/* AST genericParser.init */
<@super.init()>
[self setTreeAdaptor:[[CommonTreeAdaptor newTreeAdaptor] retain]];
>>

@genericParser.dealloc() ::= <<
/* AST genericParser.dealloc */
[self setTreeAdaptor:nil];
<@super.dealloc()>
```

```
>>

/* Add an adaptor property that knows how to build trees */
parserMemVars() ::= <<
/* AST parserMemVars */
id\<TreeAdaptor> treeAdaptor;
>>

/* Add an adaptor property that knows how to build trees */
parserProperties() ::= <<
/* AST parserProperties */
@property (retain, getter=getTreeAdaptor, setter=setTreeAdaptor:) id\<TreeAdaptor> treeAdaptor;
>>

/**
 Declaration of additional tree support methods - go in interface of parserHeaderFile() */
parserMethodsDecl() ::= <<
/* AST parserMethodsDecl */
- (id\<TreeAdaptor>) getTreeAdaptor;
- (void) setTreeAdaptor:(id\<TreeAdaptor>)theTreeAdaptor;
>>

/* Add an adaptor property that knows how to build trees */
parserSynthesize() ::= <<
/* AST parserProperties */
@synthesize treeAdaptor;
>>

/** Definition of addition tree support methods - go in implementation of genericParser() */
parserMethods() ::= <<
/* AST parserMethods */
- (id\<TreeAdaptor>) getTreeAdaptor
{
 return treeAdaptor;
}

- (void) setTreeAdaptor:(id\<TreeAdaptor>)aTreeAdaptor
{
 if (aTreeAdaptor != treeAdaptor) {
  treeAdaptor = aTreeAdaptor;
 }
}
>>

/** addition memVars for returnscopes */
@returnScopeInterface.memVars() ::= <<
/* AST returnScopeInterface.memVars */
<recognizer.ASTLabelType; null="CommonTree"> *tree;
```

```
>>

/** the interface of returnScope properties */
@returnScopeInterface.properties() ::= <<
/* AST returnScopeInterface.properties
 */
@property (retain, getter=getTree, setter=setTree:) <recognizer.ASTLabelType; null="CommonTree"> *tree;
>>

/** the interface of returnScope methodsDecl */
@returnScopeInterface.methodsDecl() ::= <<
/* AST returnScopeInterface.methodsDecl */
- (<recognizer.ASTLabelType; null="CommonTree"> *)getTree;<\n>
- (void) setTree:(<recognizer.ASTLabelType; null="CommonTree"> *)aTree;<\n>
>>

/** the implementation of returnScope synthesize */
@returnScopeImplementation.synthesize() ::= <<
/* AST returnScope.synthesize */
@synthesize tree;
>>

/** the implementation of returnScope methods */
@returnScopeImplementation.methods() ::= <<
/* AST returnScope.methods */
- (<ASTLabelType> *)getTree
{
   return tree;
}

- (void) setTree:(<ASTLabelType> *)aTree
{
   if (tree != aTree) {
       if (tree != nil) [tree release];
       if (aTree != nil) [aTree retain];
       tree = aTree;
   }
}

- (void) dealloc
{
   self.tree = nil;
   [super dealloc];
}

>>

/** Add
```

```
 a variable to track rule's return AST */
ruleDeclarations() ::= <<
/* AST ruleDeclarations */
<super.ruleDeclarations()>
<ASTLabelType> *root_0 = nil;<\n>
>>

ruleLabelDefs() ::= <<
/* AST ruleLabelDefs */
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
 ruleDescriptor.wildcardTreeListLabels]:{it | <ASTLabelType> *<it.label.text>_tree=nil;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <ASTLabelType> *<it.label.text>_tree = nil;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites:{it | RewriteRuleTokenStream *stream_<it> =
   [[RewriteRule<rewriteElementType>Stream newRewriteRule<rewriteElementType>Stream:treeAdaptor
                                                  description:@"token <it>"] retain];}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites:{it | RewriteRuleSubtreeStream *stream_<it> =
   [[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor

               description:@"rule <it>"] retain];}; separator="\n">
>>

ruleCleanUp() ::= <<
/* AST ruleCleanUp */
<super.ruleCleanUp()>
<[ruleDescriptor.allTokenRefsInAltsWithRewrites,ruleDescriptor.allRuleRefsInAltsWithRewrites]:{it |
[stream_<it> release];}; separator="\n">
<!
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(backtracking)>if ( state.backtracking == 0 ) {<\n>
<endif>
   [<prevRuleRootRef()> setTree:(<ASTLabelType> *)[treeAdaptor rulePostProcessing:root_0]];<\n>
   [treeAdaptor setTokenBoundaries:[<prevRuleRootRef()> getTree]
                             From:[<prevRuleRootRef()> getStart]
                               To:[<prevRuleRootRef()> getStop]];<\n>
<if(backtracking)>}<\n>
<endif>
<endif>
[root_0 release];
!>
>>

rewriteCodeLabelsCleanUp() ::= <<
/* AST rewriteCodeLabelsCleanUp */
<referencedTokenLabels:{it | [stream_<it> release];}; separator="\n">
<referencedTokenListLabels:{it | [stream_<it> release];}; separator="\n">
<referencedRuleLabels:{it |
 [stream_<it> release];}; separator="\n">
<referencedRuleListLabels:{it | [stream_<it> release];}; separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType> *)[[[treeAdaptor class] newEmptyTree] retain];<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex) ::= <<
<! <super.tokenRef(...)> !>
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) <endif>
  [stream_<token> addElement:<label>];<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the
 rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex) ::= <<
<! <super.tokenRef(...)> !>
<tokenRefBang(...)>
<if(backtracking)>
if ( !<actions.(actionScope).synpredgate> ) <endif>
  [stream_<token> addElement:<label>];<\n>
>>
```

```
/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
[stream_<rule.name> addElement:[<label> getTree]];
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem={[<label> getTree]},...)>
>>

/**
 ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<! <super.ruleRefRuleRoot(...)> !>
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
   [stream_<rule.name> addElement:[<label> getTree]];<\n>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem={[<label> getTree]},...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<
```

```
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list
 labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;<\n>
<rewriteCodeLabels()>
root_0 = (<ASTLabelType> *)[[[treeAdaptor class] newEmptyTree] retain];<\n>
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = (<ASTLabelType>)[treeAdaptor rulePostProcessing:root_0];
[input replaceChildren:[treeAdaptor getParent:retval.start]
                From:[treeAdaptor getChildIndex:retval.start]
                  To:[treeAdaptor getChildIndex:_last]
                With:retval.tree];
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;<\n>
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree
 = root_0;<\n>
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{it | RewriteRule<rewriteElementType>Stream *stream_<it> =
   [[RewriteRule<rewriteElementType>Stream newRewriteRule<rewriteElementType>Stream:treeAdaptor
description:@"token <it>" element:<it>] retain];};
   separator="\n"
>
<referencedTokenListLabels:{it | RewriteRule<rewriteElementType>Stream *stream_<it> =
   [[RewriteRule<rewriteElementType>Stream newRewriteRule<rewriteElementType>Stream:treeAdaptor
```

```
        description:@"token <it>" elements:list_<it>] retain];};
    separator="\n"
>
<referencedWildcardLabels:{it | RewriteRuleSubtreeStream stream_<it> =
    [[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
        description:"wildcard <it>" element:<it>] retain];};
    separator="\n"
>
<referencedWildcardListLabels:{it | RewriteRuleSubtreeStream stream_<it> =
    [[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
         descriptor:"wildcard <it>" elements:list_<it>] retain];};
    separator="\n"
>
<referencedRuleLabels:{it | RewriteRuleSubtreeStream *stream_<it> =
    [[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
        description:@"token <it>" element:<it>!=nil?[<it> getTree]:nil] retain];};
    separator="\n"
>
<referencedRuleListLabels:{it | RewriteRuleSubtreeStream *stream_<it> =
    [[RewriteRuleSubtreeStream newRewriteRuleSubtreeStream:treeAdaptor
        description:@"token <it>" elements:list_<it>] retain];};
    separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | [stream_<el> hasNext]}; separator="||">
 ) {
 <alt>
}
<referencedElementsDeep:{el | [stream_<el> reset];<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
```

```
// <fileName>:<description>
while ( <referencedElements:{el | [stream_<el> hasNext]}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | [stream_<el> reset];<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
{
if ( !(<referencedElements:{el | [stream_<el> hasNext]}; separator=" || ">) ) {
   @throw [RewriteEarlyExitException newException];
}
while ( <referencedElements:{el | [stream_<el> hasNext]}; separator=" || "> ) {
   <alt>
}
<referencedElements:{el | [stream_<el> reset];<\n>}>
}
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
 {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = nil;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
   <ASTLabelType> *root_<treeLevel> = (<ASTLabelType> *)[[[treeAdaptor class] newEmptyTree] retain];
   <root:rewriteElement()>
   <children:rewriteElement()>
```

```
  [treeAdaptor addChild:root_<treeLevel> toTree:root_<enclosingTreeLevel>];
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
// TODO: args: <args; separator=", ">
[treeAdaptor addChild:<createRewriteNodeFromElement(...)> toTree:root_<treeLevel>];<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
[treeAdaptor addChild:[stream_<label> nextNode] toTree:root_<treeLevel>];<\n>
>>

/**
 Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
[treeAdaptor addChild:[stream_<label> nextNode] toTree:root_<treeLevel>];<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:[stream_<label> nextNode]
old:root_<treeLevel>];<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<createRewriteNodeFromElement(...)>
old:root_<treeLevel>];<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
[treeAdaptor addChild:<createImaginaryNode(tokenType=token, ...)> toTree:root_<treeLevel>];<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
```

```
root_<treeLevel> = (<ASTLabelType>
 *)[treeAdaptor becomeRoot:<createImaginaryNode(tokenType=token, ...)> old:root_<treeLevel>];<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
[treeAdaptor addChild:[stream_<rule> nextTree] toTree:root_<treeLevel>];<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:(id\<Tree>)[stream_<rule> nextNode]
old:root_<treeLevel>];<\n>
>>

rewriteNodeAction(action) ::= <<
[treeAdaptor addChild:<action> toTree:root_<treeLevel>];<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<action>
 old:root_<treeLevel>];<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
[treeAdaptor addChild:[stream_<label> nextTree] toTree:root_<treeLevel>];<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
[treeAdaptor addChild:[stream_<label> nextTree] toTree:root_<treeLevel>];<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:[stream_<label> nextNode]
old:root_<treeLevel>];<\n>
```

```
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:[stream_<label> nextNode]
old:root_<treeLevel>];<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
[treeAdaptor addChild:[stream_<label> nextTree] toTree:root_<treeLevel>];<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
   [<terminalOptions.node> new<terminalOptions.node>:<tokenType> <if(args)>, <args; separator=", "><endif>]
<else>
  <if(args)>
     [[treeAdaptor createTree:<tokenType> <if(first(args))>FromToken:<first(args)><endif>
<if(first(rest(args)))>Text:<first(rest(args))><else>Text:@"<tokenType>"<endif>] retain]
  <else>
     [[treeAdaptor createTree:<tokenType> Text:@"<tokenType>"] retain]
  <endif>
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
  [<terminalOptions.node> new<terminalOptions.node>:[stream_<token> nextToken]<if(args)>, <args;
separator=", "><endif>]
<else>
  <if(args)> <! must create new node from old !>
     [[treeAdaptor createTree:<token> Text:<first(rest(args))> <args; separator=", ">] retain]
  <else>
     [stream_<token> nextNode]
  <endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/AST.stg
No license file was found, but licenses were detected in source scan.

```
* Copyright (c) 2011 Terence Parr
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY
 EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.StringTemplate;
using Antlr.StringTemplate.Language;
<if(!backtracking)>
using Hashtable = System.Collections.Hashtable;
<endif>

>>

/** Add this
 to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
private StringTemplate _st;
public StringTemplate Template { get { return _st; } set { _st = value; } }
public override string ToString() { return (Template==null) ? string.Empty : Template.ToString(); }
>>
```

```
@genericParser.members() ::= <<
<@super.members()>
protected StringTemplateGroup templateLib = new StringTemplateGroup("<name>Templates",
typeof(AngleBracketTemplateLexer) );

public StringTemplateGroup TemplateLib
{
 get { return this.templateLib; }
 set { this.templateLib = value; }
}

/// \<summary> Allows convenient multi-value initialization:
///  "new STAttrMap().Add(...).Add(...)"
/// \</summary>
protected class STAttrMap : Hashtable
{
 public STAttrMap Add(string attrName, object value)
 {
   base.Add(attrName, value);
   return this;
 }
 public STAttrMap Add(string attrName, int value)
 {
   base.Add(attrName, value);
   return this;
 }
}
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Template",...)>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
 <alts:rewriteTemplateAlt(); separator="else ">
 <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
```

```
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
((TokenRewriteStream)input.TokenStream).Replace(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start),
 retval.Template);
<else>
((TokenRewriteStream)input).Replace(
 ((IToken)retval.Start).TokenIndex,
 input.LT(-1).TokenIndex,
 retval.Template);
<endif>
>>

rewriteTemplateAlt() ::= <<
// <it.description>
<if(it.pred)>
if (<it.pred>) {
   retval.Template = <it.alt>;
}<\n>
<else>
{
   retval.Template = <it.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts)
 ::= <<
null;
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args) ::= <<
templateLib.GetInstanceOf("<name>"<if(args)>,
 new STAttrMap()<args:{a | .Add("<a.name>", <a.value>)}>
 <endif>)
>>

/** expr is a string expression that says what template to load */
```

```
rewriteIndirectTemplate(expr,args) ::= <<
templateLib.GetInstanceOf(<expr><if(args)>,
 new STAttrMap()<args:{a | .Add("<a.name>", <a.value>)}>
 <endif>)
>>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <<
new StringTemplate(templateLib, "<template>"<if(args)>,
 new STAttrMap()<args:{a | .Add("<a.name>", <a.value>)}>
 <endif>)
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/**
 An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>).SetAttribute("<attrName>",<expr>);
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(templateLib,<stringExpr>)
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/ST.stg
No license file was found, but licenses were detected in source scan.

```
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

group Perl5;

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,

   scopes, superClass, literals) ::=
<<
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>

<@imports>
<if(TREE_PARSER)>
<endif>
<if(backtracking)>
<endif>
<@end>

<docComment>
<recognizer>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="ANTLR::Runtime::Lexer")  ::= <<
package <name>;
```

```
use Carp;
use English qw( -no_match_vars ) ;
use Readonly;
use Switch;

use ANTLR::Runtime::BaseRecognizer;
use ANTLR::Runtime::DFA;
use ANTLR::Runtime::NoViableAltException;

use Moose;

extends 'ANTLR::Runtime::Lexer';

Readonly my $HIDDEN => ANTLR::Runtime::BaseRecognizer->HIDDEN;
sub HIDDEN { $HIDDEN }

use constant {
   <tokens:{ <it.name> => <it.type>, }; separator="\n">
};
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<actions.lexer.members>

sub BUILD {
   my ($self, $arg_ref) = @_;

   $self->init_dfas();
}

sub get_grammar_file_name {
   return "<fileName>";
}

<if(filterMode)>
<filteringNextToken()>
<endif>
<rules;
 separator="\n\n">

<synpreds:{p | <lexerSynpred(p)>}>

<cyclicDFAs:{dfa | has 'dfa<dfa.decisionNumber>';}; separator="\n">

sub init_dfas {
   my ($self) = @_;

   <cyclicDFAs:{dfa |
   $self->dfa<dfa.decisionNumber>(<name>::DFA<dfa.decisionNumber>->new({ recognizer => $self }));
   }; separator="\n">
   return;
}

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

no Moose;
__PACKAGE__->meta->make_immutable();
1;

>>

perlTypeInitMap ::= [
 "$":"undef",
 "@":"()",
 "%":"()",
 default:"undef"
]

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public Token nextToken() {
   while (true) {
       if ( input.LA(1)==CharStream.EOF
 ) {
           return Token.EOF_TOKEN;
       }
       token = null;
 channel = Token.DEFAULT_CHANNEL;
       tokenStartCharIndex = input.index();
       tokenStartCharPositionInLine = input.getCharPositionInLine();
       tokenStartLine = input.getLine();
 text = null;
       try {
           int m = input.mark();
           backtracking=1; <! means we won't throw slow exception !>
           failed=false;
           mTokens();
           backtracking=0;
           <! mTokens backtracks with synpred at backtracking==2
              and we set the synpredgate to allow actions at level 1. !>
           if ( failed ) {
               input.rewind(m);
```

```
            input.consume(); <! advance one char and try again !>
          }
          else {
            emit();
            return token;
          }
        }
        catch (RecognitionException re) {
          // shouldn't happen in backtracking mode, but...
          reportError(re);

        recover(re);
        }
    }
}

public void memoize(IntStream input,
  int ruleIndex,
  int ruleStartIndex)
{
if ( backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
if ( backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>

actionGate() ::= "$self->state->backtracking==0"

filteringActionGate() ::= "backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
             bitsets, inputStreamType, superClass, filterMode,
             ASTLabelType="Object", labelType, members) ::= <<
package <name>;

use English qw( -no_match_vars ) ;
use Readonly;
use Switch;
use Carp;
use ANTLR::Runtime::BitSet;

use Moose;

extends '<@superClassName><superClass><@end>';
```

```
Readonly my $token_names => [
   "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
];

use constant {
<tokens:{ <it.name>
 => <it.type>, }; separator="\n">
};

<bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                    words64=it.bits)>

<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<@members>
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>

sub BUILD {
   my ($self, $arg_ref) = @_;

<if(backtracking)>
   $self->state->rule_memo({});<\n>
<endif>
}
<@end>

sub get_token_names {
   return $token_names;
}

sub get_grammar_file_name {
   return "<fileName>";
}

<members>

<rules; separator="\n\n">

<synpreds:{p | <synpred(p)>}>

<cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = __PACKAGE__::DFA<dfa.decisionNumber>->new($self);};
separator="\n">
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

no Moose;
__PACKAGE__->meta->make_immutable();
1;
__END__
```

```
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType,
superClass="ANTLR::Runtime::Parser", labelType="ANTLR::Runtime::Token",
members={<actions.parser.members>})
 ::= <<
<genericParser(inputStreamType="ANTLR::Runtime::TokenStream", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="Object", superClass="ANTLR::Runtime::TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="TreeNodeStream", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start
 <ruleName>
sub <ruleName>_fragment {
# <ruleDescriptor.parameterScope:parameterScope(scope=it)>

<if(trace)>
   $self->traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
   eval {
       <block>
   };
   $self->traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
   if ($EVAL_ERROR) {
       croak $EVAL_ERROR;
   }
<else>
   <block>
<endif>
}
# $ANTLR end <ruleName>
>>
```

```
synpred(name) ::= <<
public final boolean <name>() {
   backtracking++;
   <@start()>
   int start = input.mark();
   try {
       <name>_fragment(); // can never throw exception
   } catch (RecognitionException re) {
       System.err.println("impossible: "+re);
   }
   boolean success = !failed;
   input.rewind(start);
   <@stop()>
   backtracking--;
   failed=false;
   return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>;
 }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>
if ($self->state->failed) {
   return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (backtracking>0) {failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
```

```
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

# $ANTLR start <ruleName>
# <fileName>:<description>
sub <ruleName>() {
   my ($self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>) = @_;
   <if(trace)>$self->traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
    <ruleDescriptor.actions.init>
   <@preamble()>
   eval {
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
   };
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   my $exception = $EVAL_ERROR;
   if (ref $exception && $exception->isa('ANTLR::Runtime::RecognitionException')) {
       $self->report_error($exception);
       $self->recover($self->input, $exception);
       $exception = undef;
   }<\n>
<endif>
<endif>
<endif>
   <if(trace)>$self->traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
   <memoize()>
   <ruleScopeCleanUp()>
   <finally>
   if ($exception) {
       croak $exception;
       #$exception->rethrow();
   }
   <@postamble()>
   return <ruleReturnValue()>;
}
# $ANTLR end <ruleName>
>>
```

```
catch(decl,action) ::= <<
catch
 (<e.decl>) {
   <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
my $retval = <returnType()>->new();
$retval->set_start($self->input->LT(1));<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
my $<a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
my $<ruleDescriptor.name>_start_index = $self->input->index();
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
   :{my $<it.label.text> = undef;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{List list_<it.label.text>=null;};
 separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|RuleReturnScope <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
```

```
   :{<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{my $<it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{List list_<it.label.text>=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
$<ruleDescriptor.singleValueReturnName>
<else>
$retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
$retval->set_stop($self->input->LT(-1));<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if
 ( backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start <ruleName>
sub m_<ruleName> {
# <ruleDescriptor.parameterScope:parameterScope(scope=it)>
   my ($self) = @_;
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleDeclarations()>
   eval {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       my $_type = <ruleName>;
       my $_channel = $self->DEFAULT_TOKEN_CHANNEL;
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       $self->state->type($_type);
       $self->state->channel($_channel);
        <(ruleDescriptor.actions.after):execAction()>
<endif>
   };
   <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
   <memoize()>

   if ($EVAL_ERROR) {
       croak $EVAL_ERROR;
   }
}
# $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
sub m_tokens {
   my ($self) = @_;
   <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
my $alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
```

```
<decision>
<@postdecision()>
<@prebranch()>
switch ($alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<
# <fileName>:<description>
my $alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch ($alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
# <fileName>:<description>
my $cnt<decisionNumber> = 0;
<decls>
<@preloop()>
LOOP<decisionNumber>:
```

```
while (1) {
   my $alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>

 switch ($alt<decisionNumber>) {
     <alts:altSwitchCase()>
     else {
         if ( $cnt<decisionNumber> >= 1 ) { last LOOP<decisionNumber> }
         <ruleBacktrackFailure()>
           my $eee =
               ANTLR::Runtime::EarlyExitException->new(<decisionNumber>, $self->input);
           <@earlyExitException()>
           croak $eee;
       }
   }
   ++$cnt<decisionNumber>;
}
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
<decls>
<@preloop()>
LOOP<decisionNumber>:
while (1) {
   my $alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch ($alt<decisionNumber>) {
     <alts:altSwitchCase()>
     else { last LOOP<decisionNumber> }
   }
}
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)?
```

```
 are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> {
   <@prealt()>
   <it>
}<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
# <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions)
 ::= <<
<if(label)>$<label> =<endif>$self->match($self->input, <token>,
$FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
```

```
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList();
list_<label>.add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
$self->match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
$self->match_range(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= $self->input->LA(1);<\n>
<else>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<endif>
if
 ( <s> ) {
   $self->input->consume();
   <postmatchCode>
<if(!LEXER)>
   $self->state->error_recovery(0);
<endif>
   <if(backtracking)>failed=false;<endif>
}
else {
   <ruleBacktrackFailure()>
   my $mse =
      ANTLR::Runtime::MismatchedSetException->new(undef, $self->input);
   <@mismatchedSetException()>
<if(LEXER)>
   $self->recover($mse);
   $mse->throw();
<else>
   $mse->throw();
   <! use following code to make it recover inline; remove throw mse;
   $self->recoverFromMismatchedSet($self->input, $mse, $FOLLOW_set_in_<ruleName><elementIndex>);
   !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
int <label>Start = getCharIndex();
$self->match(<string>); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE,
 Token.DEFAULT_CHANNEL, <label>Start, getCharIndex()-1);
<else>
$self->match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
```

```
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
$self->push_follow($FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
$<label> = $self-><rule.name>(<args; separator=", ">);<\n>
<else>
$self-><rule.name>(<args; separator=", ">);<\n>
<endif>
$self->state->_fsp($self->state->_fsp
 - 1);
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
$self->m_<rule>(<args; separator=", ">); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
$self->m_<rule.name>(<args; separator=", ">); <checkRuleBacktrackFailure()>
<endif>
```

```
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer
 */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction
 expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
   <ruleBacktrackFailure()>
   throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
```

```
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels">
else {
<if(eotPredictsAlt)>
   $alt<decisionNumber> = <eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   my $nvae =
       ANTLR::Runtime::NoViableAltException->new({
           grammar_decision_description => "<description>",
           decision_number => <decisionNumber>,
           state_number => <stateNumber>,
           input => $self->input,
       });<\n>
   <@noViableAltException()>
   croak $nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but
 this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels"><\n>
<if(eotPredictsAlt)>
<if(!edges)>
$alt<decisionNumber> = <eotPredictsAlt>; <! if no edges, don't gen ELSE !>
```

```
<else>
else {
   $alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/**
 An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "$alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k>) ) {
   <edges; separator="\n">
   else {
   <if(eotPredictsAlt)>
       $alt<decisionNumber> = <eotPredictsAlt>;
   <else>
       <ruleBacktrackFailure()>
       my $nvae =
           ANTLR::Runtime::NoViableAltException->new({
               grammar_decision_description => "<description>",
              decision_number => <decisionNumber>,
              state_number => <stateNumber>,
              input => $self->input,
           });<\n>
       <@noViableAltException()>
       croak $nvae;<\n>
   <endif>
   }
}<\n>
>>
```

```
dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k>) ) {
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
else { $alt<decisionNumber> = <eotPredictsAlt> }<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
case [<labels:{ <it> }; separator=", ">] { <targetState> }
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA
 case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
$alt<decisionNumber> = $self->dfa<decisionNumber>->predict($self->input);
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
Readonly my $DFA<dfa.decisionNumber>_eot => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedEOT; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_eof => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedEOF; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_min => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedMin; separator=", "> ]);
Readonly
```

```
 my $DFA<dfa.decisionNumber>_max => ANTLR::Runtime::DFA->unpack_rle([ <dfa.javaCompressedMax;
separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_accept => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedAccept; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_special => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedSpecial; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_transition => [
<dfa.javaCompressedTransition:{s|ANTLR::Runtime::DFA->unpack_rle([ <s; separator=", "> ])}; separator=", "> ];

{
package <name>::DFA<dfa.decisionNumber>;
use ANTLR::Runtime::Class;

use strict;
use warnings;

extends 'ANTLR::Runtime::DFA';

sub BUILD {
   my $self = shift;
   my $param_ref = __PACKAGE__->unpack_params(@_, {
       spec => [
           {
               name => 'recognizer',
               isa  => 'ANTLR::Runtime::BaseRecognizer'
           },
       ]
   });

   $self->recognizer($param_ref->{recognizer});
   $self->decision_number(<dfa.decisionNumber>);
    $self->eot($DFA<dfa.decisionNumber>_eot);
   $self->eof($DFA<dfa.decisionNumber>_eof);
   $self->min($DFA<dfa.decisionNumber>_min);
   $self->max($DFA<dfa.decisionNumber>_max);
   $self->accept($DFA<dfa.decisionNumber>_accept);
   $self->special($DFA<dfa.decisionNumber>_special);
   $self->transition($DFA<dfa.decisionNumber>_transition);
}

sub get_description {
   return "<dfa.description>";
}

<@errorMethod()>

<if(dfa.specialStateSTs)>
sub special_state_transition {
   my ($self, $param_ref) = unpack_params(@_, {
       spec => [
           {
               name => 's',
               type => SCALAR,
           },
           {
               name => 'input',
               isa  => 'ANTLR::Runtime::IntStream',
           }
       ]
   });
   my $s = $param_ref->{s};
   my $input = $param_ref->{input};

   switch ($s) {
       <dfa.specialStateSTs:{state |
       case <i0> \{ <! compressed special state numbers 0..n-1 !>
           <state>};
 separator="\n">
       }
   }

<if(backtracking)>
   if ($self->state->backtracking > 0) {
       $self->state->failed = 1;
       return -1;
   }<\n>
<endif>

   my $nvae =
       ANTLR::Runtime::NoViableAltException->new({
           grammar_decision_description => $self->get_description(),
           decision_number => <dfa.decisionNumber>,
           state_number => $s,
           input => $input,
       });<\n>
   $self->error($nvae);
   $nvae->throw();
   }<\n>
<endif>
}<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
```

```
my $input = $self->input;
my $LA<decisionNumber>_<stateNumber> = $input->LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
my $index<decisionNumber>_<stateNumber> = $input->index();
$input->rewind();<\n>
<endif>
s = -1;
<edges; separator="\nels">
<if(semPredState)> <!
 return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left> && <right>)"

orPredicates(operands) ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "$LA<decisionNumber>_<stateNumber> eq <atom>"

/**
 Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
```

```
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "$self->input->LA(<k>) eq <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
($LA<decisionNumber>_<stateNumber> ge <lower> && $LA<decisionNumber>_<stateNumber> le <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "($self->input->LA(<k>) ge
<lower> && $self->input->LA(<k>) le <upper>)"

setTest(ranges) ::= "<ranges; separator=\" || \">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected
 Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.name>_return
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
*  values.
*/
```

```
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "undef".
*/
initValue(typeName) ::= <<
<if(typeName)>
<perlTypeInitMap.(typeName)>
<else>
undef
<endif>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
my $<label.label.text> = <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/**
 Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
{
   package <returnType()>;
   use ANTLR::Runtime::Class;

   extends 'ANTLR::Runtime::<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope';

   <scope.attributes:{public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
}
<endif>
>>

parameterScope(scope) ::= <<
```

```
<scope.attributes:{$<it.name>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "$<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "$<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex)
 ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name> =<expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name> =<expr>;
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
$<scope>.<attr.name>
<else>
$<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
```

```
retval.<attr.name>
<else>
$<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr)
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>;
<else>
$<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "$<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "$<scope>->get_text()"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "((<labelType>)<scope>.start)"
ruleLabelPropertyRef_stop(scope,attr) ::= "((<labelType>)<scope>.stop)"
ruleLabelPropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)<scope>.tree)"
ruleLabelPropertyRef_text(scope,attr)
 ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
 input.getTreeAdaptor().getTokenStopIndex(<scope>.start))
<else>
substr($self->input, $<scope>->start, $<scope>->stop)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"
```

```
/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "$<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>.getText()"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.start)"
rulePropertyRef_stop(scope,attr)
 ::= "((<labelType>)retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(retval.start),
 input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "$_type"
lexerRulePropertyRef_line(scope,attr) ::= "tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "$_channel"
lexerRulePropertyRef_start(scope,attr) ::= "tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_self(scope,attr)
 ::= "$self"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"



/** How to execute an action */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
if ( backtracking==0 ) {
 <action>
}
<endif>
<else>
<action>
<endif>
>>

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
Readonly my $<name> => ANTLR::Runtime::BitSet->new({ words64 => [ <words64:{'<it>'};separator=", "> ]
});<\n>
>>

codeFileExtension() ::= ".pm"

true() ::= "1"
false() ::= "0"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Perl5/Perl5.stg
No license file was found, but licenses were detected in source scan.

```
@outputFile.imports() ::= <<
<@super.imports()>

<if(!TREE_PARSER)>
<! tree parser would already have imported !>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
<treeAdaptorType()>
 treeAdaptor = default(<treeAdaptorType()>);
CreateTreeAdaptor(ref treeAdaptor);
TreeAdaptor = treeAdaptor<if(!actions.(actionScope).treeAdaptorType)> ?? new CommonTreeAdaptor()<endif>;
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
// Implement this function in your helper file to use a custom tree adaptor
partial void CreateTreeAdaptor(ref <treeAdaptorType()> adaptor);

private <treeAdaptorType()> adaptor;

public <treeAdaptorType()> TreeAdaptor
{
 get
 {
  return adaptor;
 }

 set
 {
  this.adaptor = value;
  <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
}
>>

treeAdaptorType() ::= <<
<actions.(actionScope).treeAdaptorType; null="ITreeAdaptor">
>>

ruleReturnBaseType() ::= <%
Ast<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<ASTLabelType>, <labelType>>
%>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType>
 root_0 = default(<ASTLabelType>);<\n>
>>

ruleLabelDefs(ruleDescriptor, labelType, ASTLabelType, rewriteElementType) ::= <%
<super.ruleLabelDefs(...)>
<if(!ruleDescriptor.isSynPred)>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
 :{it|<\n><ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);}>
<ruleDescriptor.tokenListLabels:{it|<\n><ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);}>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
 :{it|<\n>RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}>
<ruleDescriptor.allRuleRefsInAltsWithRewrites
 :{it|<\n>RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");}>
<endif>
%>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same
 alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST && outerAlt && !rewriteMode && !ruleDescriptor.isSynPred)>
```

```
root_0 = (<ASTLabelType>)adaptor.Nil();
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);
>>

/**
 Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule.name>.Add(<label>.Tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
```

```
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule>.Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::= <<
<\n>{
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if (<actions.(actionScope).synpredgate>) {
<endif>
<prevRuleRootRef()>.Tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.Nil();
<alts:rewriteAlt();
 separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER&&rewriteMode)>
<prevRuleRootRef()>.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
if (<prevRuleRootRef()>.Tree != null)
 input.ReplaceChildren(adaptor.GetParent(retval.Start), adaptor.GetChildIndex(retval.Start),
adaptor.GetChildIndex(_last), retval.Tree);
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER||!rewriteMode)>
<prevRuleRootRef()>.Tree = root_0;
<endif>
<if(backtracking)>
}
<endif>
}

>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
 :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>);};
 separator="\n"
>
<referencedTokenListLabels
 :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
 separator="\n"
>
<referencedWildcardLabels
 :{it|RewriteRuleSubtreeStream
 stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
 separator="\n"
>
<referencedWildcardListLabels
 :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",list_<it>);};
 separator="\n"
>
<referencedRuleLabels
 :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.Tree:null);};
 separator="\n"
>
<referencedRuleListLabels
 :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it>",list_<it>);};
 separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
```

```
// <fileName>:<description>
if (<referencedElementsDeep:{el | stream_<el>.HasNext}; separator="||">)
{
 <alt>
}
<referencedElementsDeep:{el
 | stream_<el>.Reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if (!(<referencedElements:{el | stream_<el>.HasNext}; separator="||">))
{
 throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
 <a.alt>
}
<else>
```

```
{
 <a.alt>
}
<endif>
>>

/** For empty rewrites:
 "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <%
<@pregen()>
DebugLocation(<e.line>, <e.pos>);<\n>
<e.el>
%>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,args,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/**
 Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>
```

```
/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,args,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,elementIndex,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,elementIndex,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What
 is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<rule>.NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<rule>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.AddChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<action>, root_<treeLevel>);<\n>
>>
```

```
/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label)
 ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

createImaginaryNode(tokenType,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
%>

createRewriteNodeFromElement(token,args,terminalOptions={})
 ::= <%
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.NextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.Create(<token>, <args; separator=", ">)
<else>
stream_<token>.NextNode()
<endif>
<endif>
%>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/AST.stg
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/

parserMembers() ::= <<
 protected DebugTreeAdaptor adaptor;
 public void setTreeAdaptor(TreeAdaptor adaptor) {
<if(grammar.grammarIsRoot)>
  this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
  this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif>
  <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
 }
 public
 TreeAdaptor getTreeAdaptor() {
  return adaptor;
 }
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy =
 new DebugEventSocketProxy(this,port,<if(TREE_PARSER)>input.getTreeAdaptor()<else>adaptor<endif>);
setDebugListener(proxy);
set<inputStreamType>(new Debug<inputStreamType>(input,proxy));
try {
 proxy.handshake();
}
catch (IOException ioe) {
 reportError(ioe);
}
>>

@ctorForRootGrammar.finally() ::= <<
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
proxy.setTreeAdaptor(adap);
>>

@ctorForProfilingRootGrammar.finally() ::=<<
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
>>

@ctorForPredefinedListener.superClassRef() ::= "super(input, dbg);"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
<endif>
>>

@rewriteElement.pregen()
 ::= "dbg.location(<e.line>,<e.pos>);"
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/ASTDbg.stg

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2010 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** ANTLR's code generator.
*
*  Generate recognizers derived from grammars.  Language independence
*  achieved through the use of STGroup objects.  All output
*  strings are completely encapsulated in the group files such as Java.stg.
*  Some computations are done that are unused by a particular language.
*  This generator just computes and sets the values into the templates;
*  the templates are free
 to use or not use the information.
*
*  To make a new code generation target, define X.stg for language X
*  by copying from existing Y.stg most closely releated to your language;
*  e.g., to do CSharp.stg copy Java.stg.  The template group file has a
*  bunch of templates that are needed by the code generator.  You can add
*  a new target w/o even recompiling ANTLR itself.  The language=X option
*  in a grammar file dictates which templates get loaded/used.
*
*  Some language like C need both parser files and header files.  Java needs
```

* to have a separate file for the cyclic DFA as ANTLR generates bytecodes
* directly (which cannot be in the generated parser Java file). To facilitate
* this,
*
* cyclic can be in same file, but header, output must be searpate. recognizer
* is in outptufile.
*/

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CodeGenerator.java
No license file was found, but licenses were detected in source scan.

/** Templates for building ASTs during normal parsing.
*
* Deal with many combinations. Dimensions are:

```
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and
 !
*  invalid. There is no huge explosion of combinations.
*/

@rule.setErrorReturnValue() ::= <<
retval.Tree = (<ASTLabelType>)adaptor.ErrorNode(input, retval.Start, input.LT(-1), re);
<! System.out.WriteLine("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if (state.backtracking == 0) {<endif>
<label>_tree = <createNodeFromToken(...)>;
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
>>

/**
 ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
```

```
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
//
 large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<super.matchSet(postmatchCode={<if(backtracking)>if (<actions.(actionScope).synpredgate>)
<endif>adaptor.AddChild(root_0, <createNodeFromToken(...)>);}, ...)>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);
<endif>
<super.matchSet(postmatchCode={<if(backtracking)>if
 (<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<createNodeFromToken(...)>, root_0);}, ...)>
>>

// RULE REF AST
```

```
/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>adaptor.AddChild(root_0, <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_0);
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRefBang(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(token=[],...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
```

```
<super.wildcard(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,terminalOptions={})
 ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>)
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
adaptor.SetTokenBoundaries(retval.Tree, retval.Start, retval.Stop);
<if(backtracking)>}<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/ASTParser.stg
No license file was found, but licenses were detected in source scan.

```
*/
/** Template overrides to add debugging to normal Java output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
@outputFile.imports() ::= <<
<@super.imports()>
import org.antlr.runtime.debug.*;
import java.io.IOException;
>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
 public static final String[] ruleNames = new String[] {
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n\t\t", separator=", ">
 };<\n>
<endif>
 public static
 final boolean[] decisionCanBacktrack = new boolean[] {
  false, // invalid decision
  <grammar.decisions:{d | <d.dfa.hasSynPred; null="false">}; wrap="\n    ", separator=", ">
 };<\n>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
 public int ruleLevel = 0;
 public int getRuleLevel() { return ruleLevel; }
 public void incRuleLevel() { ruleLevel++; }
 public void decRuleLevel() { ruleLevel--; }
<if(profile)>
 <ctorForProfilingRootGrammar()>
<else>
 <ctorForRootGrammar()>
<endif>
 <ctorForPredefinedListener()>
<else><! imported grammar !>
 public int getRuleLevel() { return <grammar.delegators:{g| <g:delegateName()>}>.getRuleLevel(); }
 public void incRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.incRuleLevel(); }
 public void decRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.decRuleLevel(); }
 <ctorForDelegateGrammar()>
```

```
<endif>
<if(profile)>
 public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
  int stopIndex = getRuleMemoization(ruleIndex,
 input.index());
  ((Profiler)dbg).examineRuleMemoization(input, ruleIndex, stopIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
  return super.alreadyParsedRule(input, ruleIndex);
 }

 @Override
 public void memoize(IntStream input,
      int ruleIndex,
      int ruleStartIndex)
 {
  ((Profiler)dbg).memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
  super.memoize(input, ruleIndex, ruleStartIndex);
 }<\n>
<endif>
 protected boolean evalPredicate(boolean result, String predicate) {
  dbg.semanticPredicate(result, predicate);
  return result;
 }<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
public <name>(<inputStreamType> input) {
 this(input, DebugEventSocketProxy.DEFAULT_DEBUGGER_PORT, new RecognizerSharedState());
}
public <name>(<inputStreamType> input, int port,
 RecognizerSharedState state) {
 super(input, state);
 <parserCtorBody()>
 <createListenerAndHandshake()>
 <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
 <@finally()>
}<\n>
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
public <name>(<inputStreamType> input) {
 this(input, new Profiler(null), new RecognizerSharedState());
}
```

```
public <name>(<inputStreamType> input, DebugEventListener dbg, RecognizerSharedState state) {
 super(input, dbg, state);
 Profiler p = (Profiler)dbg;
 p.setParser(this);
 <parserCtorBody()>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state, this<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
 <@finally()>
}
<\n>
>>

/** Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
public
 <name>(<inputStreamType> input, DebugEventListener dbg, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>) {
 super(input, dbg, state);
 <parserCtorBody()>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>(input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
}<\n>
>>

ctorForPredefinedListener() ::= <<
public <name>(<inputStreamType> input, DebugEventListener dbg) {
 <@superClassRef>super(input, dbg, new RecognizerSharedState());<@end>
<if(profile)>
 Profiler p = (Profiler)dbg;
 p.setParser(this);
<endif>
 <parserCtorBody()>
 <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
 <@finally()>
}<\n>
>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy =
 new DebugEventSocketProxy(this, port, input.getTreeAdaptor());<\n>
<else>
DebugEventSocketProxy
 proxy =
 new DebugEventSocketProxy(this, port, null);<\n>
```

```
<endif>
setDebugListener(proxy);
try {
 proxy.handshake();
}
catch (IOException ioe) {
 reportError(ioe);
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

@rule.preamble() ::= <<
try { dbg.enterRule(getGrammarFileName(), "<ruleName>");
if ( getRuleLevel()==0 ) {dbg.commence();}
incRuleLevel();
dbg.location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine>);<\n>
>>

@rule.postamble() ::= <<
dbg.location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
}
finally {
 dbg.exitRule(getGrammarFileName(), "<ruleName>");
 decRuleLevel();
 if ( getRuleLevel()==0 ) {dbg.terminate();}
}<\n>
>>

@synpred.start() ::= "dbg.beginBacktrack(state.backtracking);"

@synpred.stop() ::= "dbg.endBacktrack(state.backtracking, success);"

// Common debug event triggers used by region overrides below

enterSubRule() ::=
 "try { dbg.enterSubRule(<decisionNumber>);<\n>"

exitSubRule()
 ::=
 "} finally {dbg.exitSubRule(<decisionNumber>);}<\n>"

enterDecision() ::=
 "try { dbg.enterDecision(<decisionNumber>, decisionCanBacktrack[<decisionNumber>]);<\n>"

exitDecision() ::=
 "} finally {dbg.exitDecision(<decisionNumber>);}<\n>"
```

```
enterAlt(n) ::= "dbg.enterAlt(<n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException()
 ::=
 "dbg.recognitionException(eee);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(altNum)>" // altNum is arg of altSwitchCase

@element.prematch() ::=
 "dbg.location(<e.line>,<e.pos>);" // e is arg of element

@matchSet.mismatchedSetException() ::=
 "dbg.recognitionException(mse);"

@dfaState.noViableAltException() ::= "dbg.recognitionException(nvae);"
```

```
@dfaStateSwitch.noViableAltException() ::= "dbg.recognitionException(nvae);"

dfaDecision(decisionNumber,description) ::= <<
try {
 isCyclicDecision = true;
 <super.dfaDecision(...)>
}
catch (NoViableAltException nvae) {
 dbg.recognitionException(nvae);
 throw nvae;
}
>>

@cyclicDFA.errorMethod() ::= <<
public void error(NoViableAltException nvae) {
 dbg.recognitionException(nvae);
}
>>

/** Force predicate validation to
 trigger an event */
evalPredicate(pred,description) ::= <<
evalPredicate(<pred>,"<description>")
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/Dbg.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined
 into tokenRef, ruleRef, ...
*/

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<if(!ruleDescriptor.isSynPred)>
<ASTLabelType> _first_0 = default(<ASTLabelType>);
<ASTLabelType> _last = default(<ASTLabelType>);
<endif>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel=false, treeLevel=false) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(rewriteMode)>
retval.Tree = (<ASTLabelType>)_first_0;
if (adaptor.GetParent(retval.Tree)!=null && adaptor.IsNil(adaptor.GetParent(retval.Tree)))
 retval.Tree = (<ASTLabelType>)adaptor.GetParent(retval.Tree);
<endif>
<if(backtracking)>}<endif>
<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last
 = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = default(<ASTLabelType>);
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(root.el.rule)>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>.Tree;
<else>
if (_first_<enclosingTreeLevel> == null) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down) {
 Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<enclosingTreeLevel>,
 root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}
<else>
<super.tree(...)>
<endif>
```

```
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<else>
<super.tokenRefBang(...)>
<endif>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text;
 format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
<endif>
<else>
<super.tokenRef(...)>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRef(...)>
```

```
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefAndListLabel(...)>
<endif>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text;
 format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
<else>
<super.tokenRefRuleRoot(...)>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,...)>
<else>
<super.tokenRefRuleRootAndListLabel(...)>
<endif>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if
```

```
 (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.DupTree(<label>);
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>;
<endif>
<else>
<super.wildcard(...)>
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text; format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.text)>,<terminalOptions.text;
 format="string"><endif>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSet(...)>
<endif>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<if(!ruleDescriptor.isSynPred)>
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
<else>
<super.matchRuleBlockSet(...)>
<endif>
```

```
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
<else>
<super.matchSetBang(...)>
<endif>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(!ruleDescriptor.isSynPred)>
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<if(terminalOptions.node)>
<label>_tree = new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text;
 format="string"><endif>);
<else>
<label>_tree =
(<ASTLabelType>)adaptor.DupNode(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(termin
alOptions.text)>,<terminalOptions.text; format="string"><endif>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
<else>
<super.matchSetRuleRoot(...)>
<endif>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>)<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<treeLevel>, <label>.Tree);
<else> <! rewrite mode !>
if (_first_<treeLevel> == null) _first_<treeLevel> = <label>.Tree;
```

```
<endif>
<else>
<super.ruleRef(...)>
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefAndListLabel(...)>
<endif>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_<treeLevel>);
<endif>
<else>
<super.ruleRefRuleRoot(...)>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},...)>
<else>
<super.ruleRefRuleRootAndListLabel(...)>
<endif>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>

/**
```

```
 x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
<else>
<super.ruleRefTrackAndListLabel(...)>
<endif>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
<else>
<super.ruleRefRuleRootTrack(...)>
<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
<else>
<super.ruleRefRuleRootTrackAndListLabel(...)>
<endif>
>>

/** Streams for token refs are tree nodes now; override to
*  change NextToken to NextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,args)
 ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif>stream_<token>.NextNode())
<else>
stream_<token>.NextNode()
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!ruleDescriptor.isSynPred)>
<if(!rewriteMode)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
* This code generating template and the associated C runtime was produced by:
* Jim Idle jimi|hereisanat|idle|dotgoeshere|ws.
* If it causes the destruction of the Universe, it will be pretty cool so long as
* I am in a different one at the time.

```
*/
cTypeInitMap ::= [
 "int"      : "0",              // Integers     start out being 0
 "long"      : "0",              // Longs        start out
 being 0
 "float"      : "0.0",           // Floats       start out being 0
 "double"     : "0.0",           // Doubles      start out being 0
 "ANTLR3_BOOLEAN"    : "ANTLR3_FALSE",   // Booleans     start out being Antlr C for false
 "byte"      : "0",              // Bytes        start out being 0
 "short"      : "0",              // Shorts       start out being 0
 "char"      : "0"              // Chars        start out being 0
]

leadIn(type) ::=
<<
/** \file
*  This <type> file was generated by $ANTLR version <ANTLRVersion>
*
*     -  From the grammar source file : <fileName>
*     -                            On : <generatedTimestamp>
<if(LEXER)>
*     -                 for the lexer : <name>Lexer
<endif>
<if(PARSER)>
*     -                for the parser : <name>Parser
<endif>
<if(TREE_PARSER)>
*     -           for the tree parser : <name>TreeParser
<endif>
*
* Editing it, at least manually, is not wise.
*
* C language generator and runtime by Jim Idle, jimi|hereisanat|idle|dotgoeshere|ws.
 *
*
>>

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile( LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,
            rewriteMode,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,
            generatedTimestamp,
            trace,
            scopes,
            superClass,
            literals
            ) ::=
<<
<leadIn("C source")>
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
//
 Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR
```

```
 CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<if(actions.(actionScope).header)>

/* =============================================================================
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =============================================================================
*/
<endif>

/* -----------------------------------------
* Include the ANTLR3 generated header file.
*/
#include    "<name>.h"
<actions.(actionScope).postinclude>
/* ----------------------------------------- */

<docComment>

<if(literals)>
/** String
 literals used by <name> that we must do things like MATCHS() with.
*  C will normally just lay down 8 bit characters, and you can use L"xxx" to
*  get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
*  we perform this little trick of defining the literals as arrays of UINT32
*  and passing in the address of these.
*/
<literals:{it | static ANTLR3_UCHAR lit_<i>[]  = <it>;}; separator="\n">

<endif>




/* MACROS that hide the C interface implementations from the
* generated code, which makes it a little more understandable to the human eye.
* I am very much against using C pre-processor macros for function calls and bits
* of code as you cannot see what is happening when single stepping in debuggers
* and so on. The exception (in my book at least) is for generated code, where you are
* not maintaining it, but may wish to read and understand it. If you single step it, you know that input()
* hides some indirect calls, but is always referring to the input
 stream. This is
* probably more readable than ctx->input->istream->input(snarfle0->blarg) and allows me to rejig
* the runtime interfaces without changing the generated code too often, without
* confusing the reader of the generated output, who may not wish to know the gory
* details of the interface inheritance.
*/

#define  CTX ctx

/* Aids in accessing scopes for grammar programmers
*/
#undef SCOPE_TYPE
#undef SCOPE_STACK
#undef SCOPE_TOP
#define SCOPE_TYPE(scope)   p<name>_##scope##_SCOPE
#define SCOPE_STACK(scope)  p<name>_##scope##Stack
#define SCOPE_TOP(scope)    ctx->p<name>_##scope##Top
#define SCOPE_SIZE(scope)  ctx->p<name>_##scope##Stack_limit
#define SCOPE_INSTANCE(scope, i) (ctx->SCOPE_STACK(scope)->get(ctx->SCOPE_STACK(scope),i))

<if(LEXER)>

/* Macros for accessing things in a lexer
*/
#undef    LEXER
#undef    RECOGNIZER
#undef    RULEMEMO
#undef    GETCHARINDEX
#undef    GETLINE
#undef    GETCHARPOSITIONINLINE
#undef    EMIT
#undef    EMITNEW
#undef
  MATCHC
#undef    MATCHS
#undef    MATCHRANGE
#undef    LTOKEN
#undef    HASFAILED
#undef    FAILEDFLAG
#undef    INPUT
#undef    STRSTREAM
#undef    LA
#undef    HASEXCEPTION
#undef    EXCEPTION
#undef    CONSTRUCTEX
#undef    CONSUME
#undef    LRECOVER
```

```
#undef    MARK
#undef    REWIND
#undef    REWINDLAST
#undef    BACKTRACKING
#undef  MATCHANY
#undef  MEMOIZE
#undef  HAVEPARSEDRULE
#undef  GETTEXT
#undef  INDEX
#undef  SEEK
#undef  PUSHSTREAM
#undef  POPSTREAM
#undef  SETTEXT
#undef  SETTEXT8

#define    LEXER     ctx->pLexer
#define    RECOGNIZER       LEXER->rec
#define  LEXSTATE    RECOGNIZER->state
#define  TOKSOURCE    LEXSTATE->tokSource
#define    GETCHARINDEX()   LEXER->getCharIndex(LEXER)
#define    GETLINE()    LEXER->getLine(LEXER)
#define    GETTEXT()    LEXER->getText(LEXER)
#define    GETCHARPOSITIONINLINE() LEXER->getCharPositionInLine(LEXER)
#define    EMIT()     LEXSTATE->type = _type; LEXER->emit(LEXER)
#define    EMITNEW(t)    LEXER->emitNew(LEXER,
 t)
#define    MATCHC(c)    LEXER->matchc(LEXER, c)
#define    MATCHS(s)    LEXER->matchs(LEXER, s)
#define    MATCHRANGE(c1,c2)     LEXER->matchRange(LEXER, c1, c2)
#define    MATCHANY()    LEXER->matchAny(LEXER)
#define    LTOKEN      LEXSTATE->token
#define    HASFAILED()    (LEXSTATE->failed == ANTLR3_TRUE)
#define    BACKTRACKING   LEXSTATE->backtracking
#define    FAILEDFLAG    LEXSTATE->failed
#define    INPUT     LEXER->input
#define    STRSTREAM    INPUT
#define  ISTREAM     INPUT->istream
#define  INDEX()     ISTREAM->index(ISTREAM)
#define  SEEK(n)     ISTREAM->seek(ISTREAM, n)
#define    EOF_TOKEN    &(LEXSTATE->tokSource->eofToken)
#define    HASEXCEPTION()   (LEXSTATE->error == ANTLR3_TRUE)
#define    EXCEPTION    LEXSTATE->exception
#define    CONSTRUCTEX()   RECOGNIZER->exConstruct(RECOGNIZER)
#define    LRECOVER()    LEXER->recover(LEXER)
#define    MARK()     ISTREAM->mark(ISTREAM)
#define    REWIND(m)    ISTREAM->rewind(ISTREAM, m)
#define
```

```
   REWINDLAST()   ISTREAM->rewindLast(ISTREAM)
#define  MEMOIZE(ri,si)   RECOGNIZER->memoize(RECOGNIZER, ri, si)
#define  HAVEPARSEDRULE(r)  RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define  PUSHSTREAM(str)   LEXER->pushCharStream(LEXER, str)
#define  POPSTREAM()    LEXER->popCharStream(LEXER)
#define  SETTEXT(str)   LEXSTATE->text = str
#define  SKIP()     LEXSTATE->token = &(TOKSOURCE->skipToken)
#define  USER1     LEXSTATE->user1
#define  USER2     LEXSTATE->user2
#define  USER3     LEXSTATE->user3
#define  CUSTOM     LEXSTATE->custom
#define  RULEMEMO    LEXSTATE->ruleMemo
#define  DBG      RECOGNIZER->debugger

/* If we have been told we can rely on the standard 8 bit or UTF16 input
* stream, then we can define our macros to use the direct pointers
* in the input object, which is much faster than indirect calls. This
* is really only significant to lexers with a lot of fragment rules (which
* do not place LA(1) in a temporary at the moment) and even then
* only if there is
 a lot of input (order of say 1M or so).
*/
#if defined(ANTLR3_INLINE_INPUT_8BIT) || defined(ANTLR3_INLINE_INPUT_UTF16)

# ifdef ANTLR3_INLINE_INPUT_8BIT

/* 8 bit character set */

#  define     NEXTCHAR ((pANTLR3_UINT8)(INPUT->nextChar))
#  define     DATAP ((pANTLR3_UINT8)(INPUT->data))

# else

#  define     NEXTCHAR ((pANTLR3_UINT16)(INPUT->nextChar))
#  define     DATAP ((pANTLR3_UINT16)(INPUT->data))

# endif

# define     LA(n) ((NEXTCHAR + n) > (DATAP + INPUT->sizeBuf) ? ANTLR3_CHARSTREAM_EOF :
(ANTLR3_UCHAR)(*(NEXTCHAR + n - 1)))
# define            CONSUME()                                   \\
{                                                         \\
   if        (NEXTCHAR \< (DATAP + INPUT->sizeBuf))                  \\
   {                                                     \\
      INPUT->charPositionInLine++;                                \\
      if  ((ANTLR3_UCHAR)(*NEXTCHAR) == INPUT->newlineChar)           \\
       {                                                 \\
         INPUT->line++;                                  \\
            INPUT->charPositionInLine        = 0;                        \\
            INPUT->currentLine               = (void *)(NEXTCHAR + 1); \\
        }                                                         \\
        INPUT->nextChar = (void *)(NEXTCHAR + 1);                       \\
    }                                                             \\
}

#else

// Pick up the input character by calling the input stream implementation.
//
#define     CONSUME()   INPUT->istream->consume(INPUT->istream)
#define     LA(n)       INPUT->istream->_LA(INPUT->istream, n)

#endif
<endif>

<if(PARSER)>
/* Macros for accessing things in the parser
*/

#undef     PARSER
#undef     RECOGNIZER
#undef     HAVEPARSEDRULE
#undef  MEMOIZE
#undef     INPUT
#undef     STRSTREAM
#undef     HASEXCEPTION
#undef     EXCEPTION
#undef     MATCHT
#undef
    MATCHANYT
#undef     FOLLOWSTACK
#undef     FOLLOWPUSH
#undef     FOLLOWPOP
#undef     PRECOVER
#undef     PREPORTERROR
#undef     LA
#undef     LT
#undef     CONSTRUCTEX
#undef     CONSUME
#undef     MARK
#undef     REWIND
#undef     REWINDLAST
#undef     PERRORRECOVERY
#undef     HASFAILED
#undef     FAILEDFLAG
#undef     RECOVERFROMMISMATCHEDSET
```

```
#undef    RECOVERFROMMISMATCHEDELEMENT
#undef  INDEX
#undef     ADAPTOR
#undef  SEEK
#undef    RULEMEMO
#undef  DBG

#define    PARSER    ctx->pParser
#define    RECOGNIZER    PARSER->rec
#define    PSRSTATE    RECOGNIZER->state
#define    HAVEPARSEDRULE(r)   RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define    MEMOIZE(ri,si)   RECOGNIZER->memoize(RECOGNIZER, ri, si)
#define    INPUT    PARSER->tstream
#define    STRSTREAM    INPUT
#define    ISTREAM    INPUT->istream
#define    INDEX()    ISTREAM->index(INPUT->istream)
#define    HASEXCEPTION()   (PSRSTATE->error == ANTLR3_TRUE)
#define    EXCEPTION    PSRSTATE->exception
#define
   MATCHT(t, fs)   RECOGNIZER->match(RECOGNIZER, t, fs)
#define    MATCHANYT()    RECOGNIZER->matchAny(RECOGNIZER)
#define    FOLLOWSTACK    PSRSTATE->following
#ifdef  SKIP_FOLLOW_SETS
#define    FOLLOWPUSH(x)
#define    FOLLOWPOP()
#else
#define    FOLLOWPUSH(x)   FOLLOWSTACK->push(FOLLOWSTACK, ((void *)(&(x))), NULL)
#define    FOLLOWPOP()    FOLLOWSTACK->pop(FOLLOWSTACK)
#endif
#define    PRECOVER()    RECOGNIZER->recover(RECOGNIZER)
#define    PREPORTERROR()   RECOGNIZER->reportError(RECOGNIZER)
#define    LA(n)    INPUT->istream->_LA(ISTREAM, n)
#define    LT(n)    INPUT->_LT(INPUT, n)
#define    CONSTRUCTEX()   RECOGNIZER->exConstruct(RECOGNIZER)
#define    CONSUME()    ISTREAM->consume(ISTREAM)
#define    MARK()    ISTREAM->mark(ISTREAM)
#define    REWIND(m)    ISTREAM->rewind(ISTREAM, m)
#define    REWINDLAST()   ISTREAM->rewindLast(ISTREAM)
#define    SEEK(n)    ISTREAM->seek(ISTREAM, n)
#define    PERRORRECOVERY   PSRSTATE->errorRecovery
#define
   FAILEDFLAG    PSRSTATE->failed
#define    HASFAILED()    (FAILEDFLAG == ANTLR3_TRUE)
#define    BACKTRACKING   PSRSTATE->backtracking
#define    RECOVERFROMMISMATCHEDSET(s)  RECOGNIZER->recoverFromMismatchedSet(RECOGNIZER,
s)
#define    RECOVERFROMMISMATCHEDELEMENT(e) RECOGNIZER-
>recoverFromMismatchedElement(RECOGNIZER, s)
```

```
#define     ADAPTOR                        ctx->adaptor
#define  RULEMEMO      PSRSTATE->ruleMemo
#define  DBG        RECOGNIZER->debugger

<endif>

<if(TREE_PARSER)>
/* Macros for accessing things in the parser
*/

#undef     PARSER
#undef     RECOGNIZER
#undef     HAVEPARSEDRULE
#undef     INPUT
#undef     STRSTREAM
#undef     HASEXCEPTION
#undef     EXCEPTION
#undef     MATCHT
#undef     MATCHANYT
#undef     FOLLOWSTACK
#undef     FOLLOWPUSH
#undef     FOLLOWPOP
#undef     PRECOVER
#undef     PREPORTERROR
#undef     LA
#undef     LT
#undef     CONSTRUCTEX
#undef     CONSUME
#undef     MARK
#undef     REWIND
#undef     REWINDLAST
#undef     PERRORRECOVERY
#undef
   HASFAILED
#undef     FAILEDFLAG
#undef     RECOVERFROMMISMATCHEDSET
#undef     RECOVERFROMMISMATCHEDELEMENT
#undef     BACKTRACKING
#undef      ADAPTOR
#undef     RULEMEMO
#undef  SEEK
#undef  INDEX
#undef  DBG

#define     PARSER       ctx->pTreeParser
#define     RECOGNIZER      PARSER->rec
#define  PSRSTATE      RECOGNIZER->state
#define     HAVEPARSEDRULE(r)    RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
```

```
#define    INPUT      PARSER->ctnstream
#define  ISTREAM      INPUT->tnstream->istream
#define    STRSTREAM     INPUT->tnstream
#define    HASEXCEPTION()    (PSRSTATE->error == ANTLR3_TRUE)
#define    EXCEPTION     PSRSTATE->exception
#define    MATCHT(t, fs)    RECOGNIZER->match(RECOGNIZER, t, fs)
#define    MATCHANYT()     RECOGNIZER->matchAny(RECOGNIZER)
#define    FOLLOWSTACK        PSRSTATE->following
#define    FOLLOWPUSH(x)    FOLLOWSTACK->push(FOLLOWSTACK, ((void *)(&(x))), NULL)
#define    FOLLOWPOP()     FOLLOWSTACK->pop(FOLLOWSTACK)
#define
   PRECOVER()     RECOGNIZER->recover(RECOGNIZER)
#define    PREPORTERROR()    RECOGNIZER->reportError(RECOGNIZER)
#define    LA(n)      ISTREAM->_LA(ISTREAM, n)
#define    LT(n)      INPUT->tnstream->_LT(INPUT->tnstream, n)
#define    CONSTRUCTEX()    RECOGNIZER->exConstruct(RECOGNIZER)
#define    CONSUME()     ISTREAM->consume(ISTREAM)
#define    MARK()      ISTREAM->mark(ISTREAM)
#define    REWIND(m)     ISTREAM->rewind(ISTREAM, m)
#define    REWINDLAST()    ISTREAM->rewindLast(ISTREAM)
#define    PERRORRECOVERY    PSRSTATE->errorRecovery
#define    FAILEDFLAG     PSRSTATE->failed
#define    HASFAILED()     (FAILEDFLAG == ANTLR3_TRUE)
#define    BACKTRACKING    PSRSTATE->backtracking
#define    RECOVERFROMMISMATCHEDSET(s)  RECOGNIZER->recoverFromMismatchedSet(RECOGNIZER,
s)
#define    RECOVERFROMMISMATCHEDELEMENT(e) RECOGNIZER-
>recoverFromMismatchedElement(RECOGNIZER, s)
#define    ADAPTOR                    INPUT->adaptor
#define  RULEMEMO     PSRSTATE->ruleMemo
#define  SEEK(n)      ISTREAM->seek(ISTREAM,
 n)
#define  INDEX()      ISTREAM->index(ISTREAM)
#define  DBG       RECOGNIZER->debugger



<endif>

#define  TOKTEXT(tok, txt)    tok, (pANTLR3_UINT8)txt

/* The 4 tokens defined below may well clash with your own #defines or token types. If so
* then for the present you must use different names for your defines as these are hard coded
* in the code generator. It would be better not to use such names internally, and maybe
* we can change this in a forthcoming release. I deliberately do not #undef these
* here as this will at least give you a redefined error somewhere if they clash.
*/
#define    UP    ANTLR3_TOKEN_UP
#define    DOWN    ANTLR3_TOKEN_DOWN
```

```
#define    EOR    ANTLR3_TOKEN_EOR
#define    INVALID ANTLR3_TOKEN_INVALID


/* =============================================================================
* Functions to create and destroy scopes. First come the rule scopes, followed
* by the global
 declared scopes.
*/

<rules: {r |<if(r.ruleDescriptor.ruleScope)>
<ruleAttributeScopeFuncDecl(scope=r.ruleDescriptor.ruleScope)>
<ruleAttributeScopeFuncs(scope=r.ruleDescriptor.ruleScope)>
<endif>}>

<recognizer.scopes:{it | <if(it.isDynamicGlobalScope)>
<globalAttributeScopeFuncDecl(it)>
<globalAttributeScopeFuncs(it)>
<endif>}>

/* ============================================================================= */

/* =============================================================================
* Start of recognizer
*/

<recognizer>

/* End of code
* =============================================================================
*/

>>
headerFileExtension() ::= ".h"

headerFile( LEXER,
        PARSER,
        TREE_PARSER,
        actionScope,
        actions,
        docComment,
        recognizer,
        name,
        tokens,
        tokenNames,
        rules,
        cyclicDFAs,
        bitsets,
        buildTemplate,

    buildAST,
        rewriteMode,
        profile,
        backtracking,
        synpreds,
        memoize,
        numRules,
        fileName,
        ANTLRVersion,
        generatedTimestamp,
        trace,
        scopes,
  superClass,
        literals
     ) ::=
<<
<leadIn("C header")>
<if(PARSER)>
* The parser <mainName()>
<endif>
<if(LEXER)>
* The lexer <mainName()>
<endif>
<if(TREE_PARSER)>
* The tree parser <mainName()>
<endif>
has the callable functions (rules) shown below,
* which will invoke the code for the associated rule in the source grammar
* assuming that the input stream is pointing to a token/text stream that could begin
* this rule.
*
* For instance if you call the first (topmost) rule in a parser grammar, you will
* get the results of a full parse, but calling a rule half way through the grammar will
* allow you to pass part of a full token stream to the parser, such as for syntax checking
* in editors
 and so on.
*
* The parser entry points are called indirectly (by function pointer to function) via
* a parser context typedef p<name>, which is returned from a call to <name>New().
*
<if(LEXER)>
* As this is a generated lexer, it is unlikely you will call it 'manually'. However
* the methods are provided anyway.
*
<endif>
* The methods in p<name> are  as follows:
```

```
*
* <rules:{r | <if(!r.ruleDescriptor.isSynPred)> - <headerReturnType(ruleDescriptor=r.ruleDescriptor,...)>
p<name>-><r.ruleDescriptor.name>(p<name>)<endif>}; separator="\n * ">
*
* The return type for any particular rule is of course determined by the source
* grammar file.
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1.
 Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#ifndef _<name>_H
#define _<name>_H
<actions.(actionScope).preincludes>
/* =============================================================================
* Standard antlr3 C runtime definitions
*/
#include    \<antlr3.h>

/* End of standard antlr 3 runtime definitions
* =============================================================================
*/
<actions.(actionScope).includes>

#ifdef __cplusplus
extern "C" {
#endif

// Forward declare the context typedef so that we can use it before it is
// properly defined. Delegators and delegates (from import statements) are
// interdependent and their context structures contain pointers to each other
// C only allows such things to be declared if you pre-declare the typedef.
//
typedef struct <name>_Ctx_struct
 <name>, * p<name>;

<if(recognizer.grammar.delegates)>
// Include delegate definition header files
//
<recognizer.grammar.delegates: {g|#include \<<g.recognizerName>.h>}; separator="\n">

<endif>



<actions.(actionScope).header>

#ifdef ANTLR3_WINDOWS
// Disable: Unreferenced parameter,        - Rules with parameters that are not used
//          constant conditional,        - ANTLR realizes that a prediction is always true (synpred usually)
//          initialized but unused variable      - tree rewrite variables declared but not needed
//          Unreferenced local variable       - lexer rule declares but does not always use _type
//          potentially unitialized variable used   - retval always returned from a rule
//   unreferenced local function has been removed - susually getTokenNames or freeScope, they can go without
warnigns
//
// These are only really displayed at warning level /W4 but that is the code ideal I am aiming at
// and the codegen must generate some of these
 warnings by necessity, apart from 4100, which is
// usually generated when a parser rule is given a parameter that it does not use. Mostly though
// this is a matter of orthogonality hence I disable that one.
//
#pragma warning( disable : 4100 )
#pragma warning( disable : 4101 )
#pragma warning( disable : 4127 )
#pragma warning( disable : 4189 )
#pragma warning( disable : 4505 )
#pragma warning( disable : 4701 )
#endif
<if(backtracking)>
```

```
/* ========================
* BACKTRACKING IS ENABLED
* ========================
*/
<endif>

<rules:{r |<headerReturnScope(ruleDescriptor=r.ruleDescriptor,...)>}>

<scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeDecl(it)><endif>}>
<rules:{r |<ruleAttributeScopeDecl(scope=r.ruleDescriptor.ruleScope)>}>
<if(recognizer.grammar.delegators)>
// Include delegator definition header files
//
<recognizer.grammar.delegators: {g|#include \<<g.recognizerName>.h>}; separator="\n">

<endif>

/** Context tracking structure for <mainName()>
*/
struct
 <name>_Ctx_struct
{
   /** Built in ANTLR3 context tracker contains all the generic elements
    *  required for context tracking.
    */
<if(PARSER)>
   pANTLR3_PARSER   pParser;
<endif>
<if(LEXER)>
   pANTLR3_LEXER    pLexer;
<endif>
<if(TREE_PARSER)>
   pANTLR3_TREE_PARSER     pTreeParser;
<endif>

<if(recognizer.grammar.delegates)>
 <recognizer.grammar.delegates:
        {g|p<g.recognizerName> <g:delegateName()>;}; separator="\n">
<endif>
<if(recognizer.grammar.delegators)>
 <recognizer.grammar.delegators:
        {g|p<g.recognizerName> <g:delegateName()>;}; separator="\n">
<endif>
<scopes:{it | <if(it.isDynamicGlobalScope)>
   <globalAttributeScopeDef(it)>
<endif>}; separator="\n\n">
<rules: {r |<if(r.ruleDescriptor.ruleScope)>
```

```
   <ruleAttributeScopeDef(scope=r.ruleDescriptor.ruleScope)>
<endif>}>

<if(LEXER)>
   <rules:{r | <if(!r.ruleDescriptor.isSynPred)><headerReturnType(ruleDescriptor=r.ruleDescriptor)>
(*m<r.ruleDescriptor.name>) (struct <name>_Ctx_struct
 * ctx<if(r.ruleDescriptor.parameterScope)>,
<endif><r.ruleDescriptor.parameterScope:parameterScope()>);<endif>}; separator="\n">
<endif>
<if(!LEXER)>
   <rules:{r | <headerReturnType(ruleDescriptor=r.ruleDescriptor)> (*<r.ruleDescriptor.name>) (struct
<name>_Ctx_struct * ctx<if(r.ruleDescriptor.parameterScope)>,
<endif><r.ruleDescriptor.parameterScope:parameterScope()>);}; separator="\n">
<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
   // Delegated rules
<recognizer.grammar.delegatedRules:{ruleDescriptor|
   <headerReturnType(ruleDescriptor)> (*<ruleDescriptor.name>)(struct <name>_Ctx_struct *
ctx<if(ruleDescriptor.parameterScope)>, <endif><ruleDescriptor.parameterScope:parameterScope()>);};
separator="\n">
<endif>

   const char * (*getGrammarFileName)();
   void            (*reset)  (struct <name>_Ctx_struct * ctx);
   void     (*free)   (struct <name>_Ctx_struct * ctx);
   <@members>
   <@end>
   <actions.(actionScope).context>
};

//
 Function protoypes for the constructor functions that external translation units
// such as delegators and delegates may wish to call.
//
ANTLR3_API p<name> <name>New         (<inputType()> instream<recognizer.grammar.delegators:{g|,
p<g.recognizerName> <g:delegateName()>}>);
ANTLR3_API p<name> <name>NewSSD      (<inputType()> instream,
pANTLR3_RECOGNIZER_SHARED_STATE state<recognizer.grammar.delegators:{g|, p<g.recognizerName>
<g:delegateName()>}>);
<if(!recognizer.grammar.grammarIsRoot)>
extern pANTLR3_UINT8   <recognizer.grammar.composite.rootGrammar.recognizerName>TokenNames[];
<endif>


/** Symbolic definitions of all the tokens that the <grammarType()> will work with.
* \{
*
* Antlr will define EOF, but we can't use that as it it is too common in
* in C header files and that would be confusing. There is no way to filter this out at the moment
* so we just undef it here for now. That isn't the value we get back from C recognizers
* anyway. We are looking
 for ANTLR3_TOKEN_EOF.
*/
#ifdef EOF
#undef EOF
#endif
#ifdef Tokens
#undef Tokens
#endif
<tokens:{it | #define <it.name>      <it.type>}; separator="\n">
#ifdef EOF
#undef EOF
#define EOF ANTLR3_TOKEN_EOF
#endif

#ifndef TOKENSOURCE
#define TOKENSOURCE(lxr) lxr->pLexer->rec->state->tokSource
#endif

/* End of token definitions for <name>
* =============================================================================
*/
/** \} */

#ifdef __cplusplus
}
#endif

#endif

/* END - Note:Keep extra line feed to satisfy UNIX systems */

>>

inputType() ::=<<
<if(LEXER)>
pANTLR3_INPUT_STREAM
<endif>
<if(PARSER)>
pANTLR3_COMMON_TOKEN_STREAM
<endif>
<if(TREE_PARSER)>
pANTLR3_COMMON_TREE_NODE_STREAM
<endif>
>>
```

```
grammarType() ::= <<
<if(PARSER)>
parser
<endif>
<if(LEXER)>
lexer
<endif>
<if(TREE_PARSER)>
tree parser
<endif>
>>

mainName() ::= <<
<if(PARSER)>
<name>
<endif>
<if(LEXER)>
<name>
<endif>
<if(TREE_PARSER)>
<name>
<endif>
>>

headerReturnScope(ruleDescriptor) ::= "<returnScope(...)>"

headerReturnType(ruleDescriptor)
 ::= <<
<if(LEXER)>
<if(!ruleDescriptor.isSynPred)>
void
<else>
<returnType()>
<endif>
<else>
<returnType()>
<endif>
>>

// Produce the lexer output
//
lexer(  grammar,
  name,
       tokens,
       scopes,
       rules,
       numRules,
       filterMode,
      superClass,
      labelType="pANTLR3_COMMON_TOKEN") ::= <<

<if(filterMode)>
/* Forward declare implementation function for ANTLR3_TOKEN_SOURCE interface when
* this is a filter mode lexer.
*/
static pANTLR3_COMMON_TOKEN <name>NextToken   (pANTLR3_TOKEN_SOURCE toksource);

/* Override the normal MEMOIZE and HAVEALREADYPARSED macros as this is a filtering
* lexer. In filter mode, the memoizing and backtracking are gated at BACKTRACKING > 1 rather
* than just BACKTRACKING. In some cases this might generate code akin to:
*   if (BACKTRACKING) if (BACKTRACKING > 1) memoize.
* However, I assume that the C compilers/optimizers are smart enough to work this one out
* these days -
 Jim
*/
#undef  MEMOIZE
#define  MEMOIZE(ri,si)   if (BACKTRACKING>1) { RECOGNIZER->memoize(RECOGNIZER, ri, si) }
#undef  HAVEPARSEDRULE
#define  HAVEPARSEDRULE(r)  if (BACKTRACKING>1) { RECOGNIZER->alreadyParsedRule(RECOGNIZER,
r) }
<endif>

/* Forward declare the locally static matching functions we have generated and any predicate functions.
*/
<rules:{r | static ANTLR3_INLINE <headerReturnType(ruleDescriptor=r.ruleDescriptor)>
<if(!r.ruleDescriptor.isSynPred)>m<endif><r.ruleDescriptor.name>    (p<name>
ctx<if(r.ruleDescriptor.parameterScope)>, <endif><r.ruleDescriptor.parameterScope:parameterScope()>);};
separator="\n">
static void <name>Free(p<name> ctx);

/* =========================================================================
* Lexer matching rules end.
* =========================================================================
*/

<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>

<actions.lexer.members>

static void
<name>Free  (p<name> ctx)
{
<if(memoize)>
 if (RULEMEMO
 != NULL)
 {
  RULEMEMO->free(RULEMEMO);
  RULEMEMO = NULL;
 }
<endif>
<if(grammar.directDelegates)>
 // Free the lexers that we delegated to
 // functions to. NULL the state so we only free it once.
 //
 <grammar.directDelegates:
        {g|ctx-><g:delegateName()>->pLexer->rec->state = NULL;
        ctx-><g:delegateName()>->free(ctx-><g:delegateName()>);}; separator="\n">
<endif>
   LEXER->free(LEXER);

   ANTLR3_FREE(ctx);
}

static void
<name>Reset (p<name> ctx)
{
   RECOGNIZER->reset(RECOGNIZER);
}

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
static const char * getGrammarFileName()
{
 return fileName;
}

<if(filterMode)>
   <filteringNextToken()>
<endif>

/** \brief Create a new lexer called <name>
*
* \param[in]    instream Pointer to an initialized input stream
* \return
*     - Success p<name> initialized
 for the lex start
*     - Fail NULL
*/
ANTLR3_API p<name> <name>New
(<inputType()> instream<grammar.delegators:{g|, p<g.recognizerName> <g:delegateName()>}>)
```

```
{
 // See if we can create a new lexer with the standard constructor
 //
 return <name>NewSSD(instream, NULL<grammar.delegators:{g|, <g:delegateName()>}>);
}

/** \brief Create a new lexer called <name>
*
* \param[in]    instream Pointer to an initialized input stream
* \param[state] state Previously created shared recognizer stat
* \return
*     - Success p<name> initialized for the lex start
*     - Fail NULL
*/
ANTLR3_API p<name> <name>NewSSD
(pANTLR3_INPUT_STREAM instream, pANTLR3_RECOGNIZER_SHARED_STATE
state<grammar.delegators:{g|, p<g.recognizerName> <g:delegateName()>}>)
{
   p<name> ctx; // Context structure we will build and return

   ctx = (p<name>) ANTLR3_CALLOC(1, sizeof(<name>));

   if  (ctx == NULL)
   {
       // Failed to allocate memory for lexer context
       return  NULL;
   }

   /*
 -------------------------------------------------------------------
    * Memory for basic structure is allocated, now to fill in
    * in base ANTLR3 structures. We initialize the function pointers
    * for the standard ANTLR3 lexer function set, but upon return
    * from here, the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here so you can get a sense
    * of what goes where.
    */

   /* Create a base lexer, using the supplied input stream
    */
   ctx->pLexer = antlr3LexerNewStream(ANTLR3_SIZE_HINT, instream, state);

   /* Check that we allocated the memory correctly
    */
   if (ctx->pLexer == NULL)
   {
```

```
  ANTLR3_FREE(ctx);
  return  NULL;
   }
<if(memoize)>
<if(grammar.grammarIsRoot)>
   // Create a LIST for recording rule memos.
   //
   ctx->pLexer->rec->ruleMemo    = antlr3IntTrieNew(15); /* 16 bit depth is enough for 32768 rules! */
<endif>
<endif>

    /* Install the implementation of our <name> interface
    */
   <rules:{r | <if(!r.ruleDescriptor.isSynPred)>ctx->m<r.ruleDescriptor.name> =
m<r.ruleDescriptor.name>;<endif>}; separator="\n">

   /** When the nextToken() call is made to this lexer's pANTLR3_TOKEN_SOURCE
    *  it will call mTokens() in this generated code, and will pass it the ctx
    * pointer of this lexer, not the context of the base lexer, so store that now.
    */
   ctx->pLexer->ctx     = ctx;

   /**Install the token matching function
    */
   ctx->pLexer->mTokens = (void (*) (void *))(mTokens);

   ctx->getGrammarFileName = getGrammarFileName;
   ctx->free  = <name>Free;
   ctx->reset          = <name>Reset;

<if(grammar.directDelegates)>
 // Initialize the lexers that we are going to delegate some
 // functions to.
 //
 <grammar.directDelegates:
       {g|ctx-><g:delegateName()> = <g.recognizerName>NewSSD(instream, ctx->pLexer->rec->state,
ctx<grammar.delegators:{g|, <g:delegateName()>}>);};
 separator="\n">
<endif>
<if(grammar.delegators)>
 // Install the pointers back to lexers that will delegate us to perform certain functions
 // for them.
 //
 <grammar.delegators:
       {g|ctx-><g:delegateName()>   = <g:delegateName()>;}; separator="\n">
<endif>
<if(filterMode)>
   /* We have filter mode turned on, so install the filtering nextToken function
```

```
     */
    ctx->pLexer->rec->state->tokSource->nextToken = <name>NextToken;
<endif>
  <actions.lexer.apifuncs>

    /* Return the newly built lexer to the caller
     */
    return  ctx;
}
<if(cyclicDFAs)>

/* ==================================================================================
* DFA tables for the lexer
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* ==================================================================================
* End of DFA tables for the lexer
*/
<endif>

/* ==================================================================================
* Functions to match the lexer grammar defined tokens from
 the input stream
*/

<rules; separator="\n\n">

/* ==================================================================================
* Lexer matching rules end.
* ==================================================================================
*/
<if(synpreds)>

/* ==================================================================================
* Lexer syntactic predicates
*/
<synpreds:{p | <lexerSynpred(predname=p)>}>
/* ==================================================================================
* Lexer syntactic predicates end.
* ==================================================================================
*/
<endif>

/* End of Lexer code
* ======================================================
* ======================================================
*/
```

```
>>


filteringNextToken() ::= <<
/** An override of the lexer's nextToken() method that backtracks over mTokens() looking
*  for matches in lexer filterMode.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  BACKTRACKING
 needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at BACKTRACKING==1.
*/
static pANTLR3_COMMON_TOKEN
<name>NextToken(pANTLR3_TOKEN_SOURCE toksource)
{
   pANTLR3_LEXER   lexer;
 pANTLR3_RECOGNIZER_SHARED_STATE state;

   lexer   = (pANTLR3_LEXER)(toksource->super);
   state = lexer->rec->state;

   /* Get rid of any previous token (token factory takes care of
    * any deallocation when this token is finally used up.
    */
   state  ->token     = NULL;
   state  ->error     = ANTLR3_FALSE;     /* Start out without an exception */
   state  ->failed    = ANTLR3_FALSE;

   /* Record the start of the token in our input stream.
    */
   state->tokenStartCharIndex   = lexer->input->istream->index(lexer->input->istream);
   state->tokenStartCharPositionInLine = lexer->input->getCharPositionInLine(lexer->input);
   state->tokenStartLine    = lexer->input->getLine(lexer->input);
   state->text       = NULL;

   /*
 Now call the matching rules and see if we can generate a new token
    */
   for (;;)
   {
  if  (lexer->input->istream->_LA(lexer->input->istream, 1) == ANTLR3_CHARSTREAM_EOF)
  {
   /* Reached the end of the stream, nothing more to do.
    */
   pANTLR3_COMMON_TOKEN    teof = &(toksource->eofToken);

   teof->setStartIndex (teof, lexer->getCharIndex(lexer));
   teof->setStopIndex  (teof, lexer->getCharIndex(lexer));
   teof->setLine  (teof, lexer->getLine(lexer));
   return  teof;
  }

  state->token  = NULL;
  state->error  = ANTLR3_FALSE;     /* Start out without an exception */

  {
   ANTLR3_MARKER   m;

   m      = lexer->input->istream->mark(lexer->input->istream);
   state->backtracking  = 1;    /* No exceptions */
   state->failed   = ANTLR3_FALSE;

   /* Call the generated lexer, see if it can get a new token together.
    */
   lexer->mTokens(lexer->ctx);
     state->backtracking = 0;

     <! mTokens backtracks with synpred at BACKTRACKING==2
    and we set the synpredgate
 to allow actions at level 1. !>

   if (state->failed == ANTLR3_TRUE)
   {
    lexer->input->istream->rewind(lexer->input->istream, m);
    lexer->input->istream->consume(lexer->input->istream); <! advance one char and try again !>
   }
   else
   {
    lexer->emit(lexer);     /* Assemble the token and emit it to the stream */
    return state->token;
   }
  }
   }
}
>>

actionGate() ::= "BACKTRACKING==0"

filteringActionGate() ::= "BACKTRACKING==1"

/** How to generate a parser */
genericParser(  grammar,
    name,
               scopes,
               tokens,
               tokenNames,
               rules,
            numRules,
            bitsets,
            inputStreamType,
            superClass,
            labelType,
    members,
    rewriteElementType, filterMode,
            ASTLabelType="pANTLR3_BASE_TREE"
           ) ::= <<


<if(grammar.grammarIsRoot)>
/** \brief Table of all token names in symbolic order, mainly used for
*
  error reporting.
*/
pANTLR3_UINT8   <name>TokenNames[<length(tokenNames)>+4]
    = {
      (pANTLR3_UINT8) "\<invalid>",       /* String to print to indicate an invalid token */
      (pANTLR3_UINT8) "\<EOR>",
      (pANTLR3_UINT8) "\<DOWN>",
      (pANTLR3_UINT8) "\<UP>",
      <tokenNames:{it |(pANTLR3_UINT8) <it>}; separator=",\n">
     };
<endif>

  <@members>

  <@end>
<rules:{r |<ruleAttributeScopeFuncMacro(scope=r.ruleDescriptor.ruleScope)>}>
<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScopeFuncMacro(it)><endif>}>

// Forward declare the locally static matching functions we have generated.
//
<rules:{r | static <headerReturnType(ruleDescriptor=r.ruleDescriptor)> <r.ruleDescriptor.name>    (p<name>
ctx<if(r.ruleDescriptor.parameterScope)>, <endif><r.ruleDescriptor.parameterScope:parameterScope()>);};
separator="\n">
static void <name>Free(p<name> ctx);
static void     <name>Reset (p<name> ctx);

<if(!LEXER)>
<! generate rule/method definitions for
 imported rules so they
  appear to be defined in this recognizer. !>
<if(recognizer.grammar.delegatedRules)>
// Delegated rules
//
<recognizer.grammar.delegatedRules:{ruleDescriptor|static <headerReturnType(ruleDescriptor)>
```

```
<ruleDescriptor.name>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>);}; separator="\n">

<endif>
<endif>

/* For use in tree output where we are accumulating rule labels via label += ruleRef
* we need a function that knows how to free a return scope when the list is destroyed.
* We cannot just use ANTLR3_FREE because in debug tracking mode, this is a macro.
*/
static void ANTLR3_CDECL freeScope(void * scope)
{
   ANTLR3_FREE(scope);
}

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
static const char * getGrammarFileName()
{
 return fileName;
}
/** \brief Create a
 new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
ANTLR3_API p<name>
<name>New   (<inputStreamType> instream<grammar.delegators:{g|, p<g.recognizerName>
<g:delegateName()>}>)
{
 // See if we can create a new parser with the standard constructor
 //
 return <name>NewSSD(instream, NULL<grammar.delegators:{g|, <g:delegateName()>}>);
}

/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
ANTLR3_API p<name>
```

```
<name>NewSSD   (<inputStreamType> instream, pANTLR3_RECOGNIZER_SHARED_STATE
state<grammar.delegators:{g|, p<g.recognizerName> <g:delegateName()>}>)
{
   p<name> ctx;     /* Context structure we will build and return   */

   ctx = (p<name>) ANTLR3_CALLOC(1, sizeof(<name>));

   if (ctx == NULL)
   {
  // Failed to allocate memory
 for parser context
  //
       return  NULL;
   }

   /* -------------------------------------------------------------------
    * Memory for basic structure is allocated, now to fill in
    * the base ANTLR3 structures. We initialize the function pointers
    * for the standard ANTLR3 parser function set, but upon return
    * from here, the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here, in order that you can get a sense
    * of what goes where.
    */

<if(PARSER)>
   /* Create a base parser/recognizer, using the supplied token stream
    */
   ctx->pParser     = antlr3ParserNewStream(ANTLR3_SIZE_HINT, instream->tstream, state);
<endif>
<if(TREE_PARSER)>
   /* Create a base Tree parser/recognizer, using the supplied tree node stream
    */
   ctx->pTreeParser  = antlr3TreeParserNewStream(ANTLR3_SIZE_HINT, instream, state);
<endif>

   /* Install the implementation
 of our <name> interface
    */
   <rules:{r | ctx-><r.ruleDescriptor.name> = <r.ruleDescriptor.name>;}; separator="\n">
<if(grammar.delegatedRules)>
 // Install the delegated methods so that they appear to be a part of this
 // parser
 //
   <grammar.delegatedRules:{ruleDescriptor | ctx-><ruleDescriptor.name> = <ruleDescriptor.name>;};
separator="\n">
<endif>
```

```
   ctx->free   = <name>Free;
   ctx->reset   = <name>Reset;
   ctx->getGrammarFileName = getGrammarFileName;

   /* Install the scope pushing methods.
    */
   <rules: {r |<if(r.ruleDescriptor.ruleScope)>
<ruleAttributeScope(scope=r.ruleDescriptor.ruleScope)><\n>
<endif>}>
   <recognizer.scopes:{it |<if(it.isDynamicGlobalScope)>
<globalAttributeScope(it)><\n>
<endif>}>
   <@apifuncs>

   <@end>
<if(grammar.directDelegates)>
 // Initialize the parsers that we are going to delegate some
 // functions to.
 //
 <grammar.directDelegates:
       {g|ctx-><g:delegateName()> = <g.recognizerName>NewSSD(instream, PSRSTATE,
 ctx<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
<endif>
<if(grammar.delegators)>
 // Install the pointers back to parsers that will delegate us to perform certain functions
 // for them.
 //
 <grammar.delegators:
       {g|ctx-><g:delegateName()>   = <g:delegateName()>;}; separator="\n">
<endif>
   <actions.parser.apifuncs>
   <actions.treeparser.apifuncs>
<if(memoize)>
<if(grammar.grammarIsRoot)>
   /* Create a LIST for recording rule memos.
    */
    RULEMEMO    = antlr3IntTrieNew(15); /* 16 bit depth is enough for 32768 rules! */<\n>
<endif>
<endif>
   /* Install the token table
    */
   PSRSTATE->tokenNames   = <grammar.composite.rootGrammar.recognizerName>TokenNames;

   <@debugStuff()>

   /* Return the newly built parser to the caller
    */
   return  ctx;
}

static void
<name>Reset (p<name> ctx)
{
   RECOGNIZER->reset(RECOGNIZER);
}

/** Free the parser resources
*/
static void
<name>Free(p<name> ctx)
{
   /* Free any scope memory
     */
   <rules: {r
|<if(r.ruleDescriptor.ruleScope)><ruleAttributeScopeFree(scope=r.ruleDescriptor.ruleScope)><\n><endif>}>
   <recognizer.scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScopeFree(it)><\n><endif>}>

   <@cleanup>
   <@end>
<if(grammar.directDelegates)>
 // Free the parsers that we delegated to
 // functions to.NULL the state so we only free it once.
 //
 <grammar.directDelegates:
       {g| ctx-><g:delegateName()>-><if(TREE_PARSER)>pTreeParser<else>pParser<endif>->rec->state = NULL;
       ctx-><g:delegateName()>->free(ctx-><g:delegateName()>);}; separator="\n">
<endif>
<if(memoize)>
<if(grammar.grammarIsRoot)>
 if (RULEMEMO != NULL)
 {
  RULEMEMO->free(RULEMEMO);
  RULEMEMO = NULL;
 }
<endif>
<endif>
 // Free this parser
 //
<if(TREE_PARSER)>
   ctx->pTreeParser->free(ctx->pTreeParser);<\n>
<else>
   ctx->pParser->free(ctx->pParser);<\n>
<endif>

   ANTLR3_FREE(ctx);
   /* Everything is released, so we can return
    */
   return;
}

/**
 Return token names used by this <grammarType()>
*
* The returned pointer is used as an index into the token names table (using the token
* number as the index).
*
* \return Pointer to first char * in the table.
*/
static pANTLR3_UINT8    *getTokenNames()
{
       return <grammar.composite.rootGrammar.recognizerName>TokenNames;
}

   <members>

/* Declare the bitsets
*/
<bitsets:{it | <bitsetDeclare(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                 words64=it.bits)>}>


<if(cyclicDFAs)>

/* =========================================================================
* DFA tables for the parser
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =========================================================================
* End of DFA tables for the parser
*/
<endif>

/* ==============================================
* Parsing rules
*/
<rules; separator="\n\n">
<if(grammar.delegatedRules)>
 // Delegated methods that appear to be a part of this
 //
 parser
 //
<grammar.delegatedRules:{ruleDescriptor|
   <returnType()> <ruleDescriptor.name>(p<name> ctx<if(ruleDescriptor.parameterScope.attributes)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>)
   \{
       <if(ruleDescriptor.hasReturnValue)>return <endif>ctx-><ruleDescriptor.grammar:delegateName()>-
><ruleDescriptor.name>(ctx-
><ruleDescriptor.grammar:delegateName()><if(ruleDescriptor.parameterScope.attributes)>,
<endif><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", ">);
 \}}; separator="\n">

<endif>
/* End of parsing rules
* ==============================================
*/

/* ==============================================
* Syntactic predicates
*/
<synpreds:{p | <synpred(predname=p)>}>
/* End of syntactic predicates
* ==============================================
*/






>>

parser( grammar,
  name,
  scopes,
  tokens,
  tokenNames,
  rules,
  numRules,
  bitsets,
  ASTLabelType,
  superClass="Parser",
  labelType="pANTLR3_COMMON_TOKEN",
  members={<actions.parser.members>}
  )
 ::= <<
<genericParser(inputStreamType="pANTLR3_COMMON_TOKEN_STREAM", rewriteElementType="TOKEN",
...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
```

```
*/
treeParser( grammar,
   name,
   scopes,
   tokens,
   tokenNames,
   globalAction,
   rules,
   numRules,
   bitsets,
   filterMode,
   labelType={<ASTLabelType>},
   ASTLabelType="pANTLR3_BASE_TREE",
   superClass="TreeParser",
   members={<actions.treeparser.members>}
   ) ::= <<
<genericParser(inputStreamType="pANTLR3_COMMON_TREE_NODE_STREAM",
rewriteElementType="NODE", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule,
 they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
static void <ruleName>_fragment(p<name> ctx <ruleDescriptor.parameterScope:parameterScope()>)
{
 <ruleLabelDefs()>
 <ruleLabelInitializations()>
<if(trace)>
   ANTLR3_PRINTF("enter <ruleName> %d failed = %d, backtracking = %d\\n",LT(1),failed,BACKTRACKING);
   <block>
   ANTLR3_PRINTF("exit <ruleName> %d, failed = %d, backtracking = %d\\n",LT(1),failed,BACKTRACKING);

<else>
   <block>
<endif>
<ruleCleanUp()>
}
// $ANTLR end <ruleName>
>>
```

```
synpred(predname) ::= <<
static ANTLR3_BOOLEAN <predname>(p<name> ctx)
{
   ANTLR3_MARKER   start;
   ANTLR3_BOOLEAN  success;

   BACKTRACKING++;
   <@start()>
   start = MARK();
   <predname>_fragment(ctx);     // can never throw exception
   success = !(FAILEDFLAG);
   REWIND(start);
   <@stop()>
   BACKTRACKING--;
   FAILEDFLAG = ANTLR3_FALSE;
   return success;
}<\n>
>>

lexerSynpred(predname)
 ::= <<
<synpred(predname)>
>>

ruleMemoization(rname) ::= <<
<if(memoize)>
if ( (BACKTRACKING>0) && (HAVEPARSEDRULE(<ruleDescriptor.index>)) )
{
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
 retval.start = 0;<\n>
<endif>
<endif>
   <(ruleDescriptor.actions.after):execAfter()>
   <finalCode(finalBlock=finally)>
<if(!ruleDescriptor.isSynPred)>
   <scopeClean()><\n>
<endif>
   return <ruleReturnValue()>;
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
if  (HASEXCEPTION())
{
   goto rule<ruleDescriptor.name>Ex;
```

```
}
<if(backtracking)>
if (HASFAILED())
{
   <scopeClean()>
   <@debugClean()>
   return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if (BACKTRACKING>0)
{
   FAILEDFLAG = <true_value()>;
   <scopeClean()>
   return <ruleReturnValue()>;
}
<endif>
>>

/** How to generate
 code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
/**
* $ANTLR start <ruleName>
* <fileName>:<description>
*/
static <returnType()>
<ruleName>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>)
{
   <if(trace)>ANTLR3_PRINTF("enter <ruleName> %s failed=%d, backtracking=%d\n", LT(1),
BACKTRACKING);<endif>
   <ruleDeclarations()>
   <ruleDescriptor.actions.declarations>
   <ruleLabelDefs()>
   <ruleInitializations()>
   <ruleDescriptor.actions.init>
   <ruleMemoization(rname=ruleName)>
   <ruleLabelInitializations()>
   <@preamble()>
   {
       <block>
   }

   <ruleCleanUp()>
<if(exceptions)>
   if (HASEXCEPTION())
   {
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
   }
   else
   {
 <(ruleDescriptor.actions.after):execAfter()>
   }
<else>
   <if(!emptyRule)>
        <if(actions.(actionScope).rulecatch)>
           <actions.(actionScope).rulecatch>
       <else>
           if (HASEXCEPTION())
           {
               PREPORTERROR();
               PRECOVER();
               <@setErrorReturnValue()>
           }
           <if(ruleDescriptor.actions.after)>
           else
           {
               <(ruleDescriptor.actions.after):execAfter()>
           }<\n>
           <endif>
       <endif>
   <endif>
<endif>

   <if(trace)>ANTLR3_PRINTF("exit <ruleName> %d failed=%s backtracking=%s\n", LT(1), failed,
BACKTRACKING);<endif>
   <memoize()>
<if(finally)>
   <finalCode(finalBlock=finally)>
<endif>
   <scopeClean()>
   <@postamble()>
   return <ruleReturnValue()>;
}
/* $ANTLR end <ruleName> */
>>

finalCode(finalBlock) ::= <<
{
   <finalBlock>
}

>>

catch(decl,action) ::= <<
/* catch(decl,action)
*/
{
   <e.action>
}

>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()>
 retval;<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name>;
}>
<endif>
<if(memoize)>
ANTLR3_MARKER <ruleDescriptor.name>_StartIndex;
<endif>
>>

ruleInitializations() ::= <<
/* Initialize rule variables
*/
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.returnScope.attributes:{ a |
<if(a.initValue)>retval.<a.name> = <a.initValue>;<endif>
}>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<if(a.initValue)><a.name> = <a.initValue>;<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex = INDEX();<\n>
<endif>
<ruleDescriptor.useScopes:{it |<scopeTop(it)> = <scopePush(it)>;}; separator="\n">
<ruleDescriptor.ruleScope:{it |<scopeTop(it.name)> = <scopePush(it.name)>;}; separator="\n">
>>

ruleLabelDefs() ::= <<
```

```
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
   :{it |<labelType>    <it.label.text>;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{it |pANTLR3_VECTOR    list_<it.label.text>;};
 separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
   :ruleLabelDef(); separator="\n"
>
>>

ruleLabelInitializations() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
   :{it |<it.label.text>       = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{it |list_<it.label.text>     = NULL;}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
   :ruleLabelInitVal(); separator="\n"
>
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
retval.start = LT(1); retval.stop = retval.start;<\n>
<endif>
<endif>
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |<labelType> <it.label.text>;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |ANTLR3_UINT32 <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
    :{it |pANTLR3_INT_TRIE list_<it.label.text>;}; separator="\n"
>
>>

lexerRuleLabelInit() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |<it.label.text> = NULL;}; separator="\n"
```

```
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{it |list_<it.label.text> = antlr3IntTrieNew(31);}; separator="\n"
>
>>

lexerRuleLabelFree() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{it |list_<it.label.text>->free(list_<it.label.text>);}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

memoize()
 ::= <<
<if(memoize)>
<if(backtracking)>
if ( BACKTRACKING>0 ) { MEMOIZE(<ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

ruleCleanUp() ::= <<

// This is where rules clean up and exit
//
goto rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex: ;
```

```
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
<if(!ruleDescriptor.isSynPred)>
retval.stop = LT(-1);<\n>
<endif>
<endif>
<endif>
>>

scopeClean() ::= <<
<ruleDescriptor.useScopes:{it |<scopePop(it)>}; separator="\n">
<ruleDescriptor.ruleScope:{it |<scopePop(it.name)>}; separator="\n">

>>
/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules, which do not produce tokens.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
//   Comes from: <block.description>
/** \brief Lexer rule generated by ANTLR3
*
* $ANTLR start <ruleName>
*
* Looks to match the characters the constitute the
 token <ruleName>
* from the attached input stream.
*
*
* \remark
*  - lexer->error == ANTLR3_TRUE if an exception was thrown.
*/
static ANTLR3_INLINE
void m<ruleName>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope()>)
{
 ANTLR3_UINT32 _type;
   <ruleDeclarations()>
   <ruleDescriptor.actions.declarations>
   <lexerRuleLabelDefs()>
   <if(trace)>System.out.println("enter <ruleName> '"+(char)LA(1)+"'
line="+GETLINE()+":"+GETCHARPOSITIONINLINE()+" failed="+failed+"
backtracking="+BACKTRACKING);<endif>

<if(nakedBlock)>
   <ruleMemoization(rname=ruleName)>
   <lexerRuleLabelInit()>
   <ruleDescriptor.actions.init>
   <block><\n>
<else>
   <ruleMemoization(rname=ruleName)>
   <lexerRuleLabelInit()>
   _type     = <ruleName>;

   <ruleDescriptor.actions.init>

   <block>
 LEXSTATE->type = _type;
<endif>
   <if(trace)> ANTLR3_FPRINTF(stderr, "exit <ruleName> '%c' line=%d:%d failed = %d, backtracking
=%d\n",LA(1),GETLINE(),GETCHARPOSITIONINLINE(),failed,BACKTRACKING);<endif>
    <ruleCleanUp()>
   <lexerRuleLabelFree()>
   <(ruleDescriptor.actions.after):execAfter()>
   <memoize>
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
/** This is the entry point in to the lexer from an object that
*  wants to generate the next token, such as a pCOMMON_TOKEN_STREAM
*/
static void
mTokens(p<name> ctx)
{
   <block><\n>

   goto ruleTokensEx; /* Prevent compiler warnings */
ruleTokensEx: ;
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<

// <fileName>:<description>
{
   int alt<decisionNumber>=<maxAlt>;
   <decls>
   <@predecision()>
   <decision>
   <@postdecision()>
   <@prebranch()>
   switch (alt<decisionNumber>)
    {
 <alts:{a | <altSwitchCase(i,a)>}>
    }
   <@postbranch()>
}
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
{
   //  <fileName>:<description>

   ANTLR3_UINT32 alt<decisionNumber>;

   alt<decisionNumber>=<maxAlt>;

   <decls>
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
   {
 <alts:{a | <altSwitchCase(i,a)>}>
   }
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>
```

```
/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n)
 ::= <<
// <fileName>:<description>
{
   int cnt<decisionNumber>=0;
   <decls>
   <@preloop()>

   for (;;)
   {
       int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch (alt<decisionNumber>)
 {
     <alts:{a | <altSwitchCase(i,a)>}>
     default:

  if ( cnt<decisionNumber> >= 1 )
  {
      goto loop<decisionNumber>;
  }
  <ruleBacktrackFailure()>
  <earlyExitEx()>
  <@earlyExitException()>
  goto rule<ruleDescriptor.name>Ex;
 }
 cnt<decisionNumber>++;
   }
   loop<decisionNumber>: ; /* Jump to here if this rule does not match */
   <@postloop()>
}
>>

earlyExitEx() ::= <<
/* mismatchedSetEx()
*/
CONSTRUCTEX();
EXCEPTION->type = ANTLR3_EARLY_EXIT_EXCEPTION;
EXCEPTION->name = (void *)ANTLR3_EARLY_EXIT_NAME;
<\n>
>>
positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
```

```
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<

// <fileName>:<description>
<decls>

<@preloop()>
for (;;)
{
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
   {
 <alts:{a | <altSwitchCase(i,a)>}>
 default:
     goto loop<decisionNumber>; /* break out of the loop */
     break;
   }
}
loop<decisionNumber>: ; /* Jump out to here if this rule does not match */
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by antlr before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
   <@prealt()>
    <alt>
   break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
```

```
{
   <@declarations()>
   <@initializations()>
   <elements:element()>
   <rew>
   <@cleanup()>
}
>>

// E L E M E N T S
/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label> = (<labelType>)<endif> MATCHT(<token>,
&FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == NULL)
{
   list_<label>=ctx->vectors->newVector(ctx->vectors);
}
list_<label>->add(list_<label>,
 <elem>, NULL);
>>



/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = LA(1);<\n>
```

```
<endif>
MATCHC(<char>);
<checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHRANGE(<a>, <b>);
<checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= LA(1);<\n>
<else>
<label>=(<labelType>)LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
  CONSUME();
  <postmatchCode>
<if(!LEXER)>
  PERRORRECOVERY=ANTLR3_FALSE;
<endif>
  <if(backtracking)>FAILEDFLAG=ANTLR3_FALSE;<\n><endif>
}
else
{
  <ruleBacktrackFailure()>
  <mismatchedSetEx()>
  <@mismatchedSetException()>
<if(LEXER)>
  LRECOVER();
<else>
  RECOVERFROMMISMATCHEDSET(&FOLLOW_set_in_<ruleName><elementIndex>);
<endif>

  goto rule<ruleDescriptor.name>Ex;
}<\n>
>>

mismatchedSetEx() ::= <<
```

```
CONSTRUCTEX();
EXCEPTION->type         = ANTLR3_MISMATCHED_SET_EXCEPTION;
EXCEPTION->name         = (void *)ANTLR3_MISMATCHED_SET_NAME;
<if(PARSER)>
EXCEPTION->expectingSet = &FOLLOW_set_in_<ruleName><elementIndex>;
<endif>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
<label>Start = GETCHARINDEX();
MATCHS(<string>);
<checkRuleBacktrackFailure()>
<label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
<label>->setType(<label>, ANTLR3_TOKEN_INVALID);
<label>->setStartIndex(<label>, <label>Start);
<label>->setStopIndex(<label>, GETCHARINDEX()-1);
<label>->input = INPUT->tnstream->istream;
<else>
MATCHS(<string>);
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=(<labelType>)LT(1);<\n>
<endif>
MATCHANYT();
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions)
 ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
```

```
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHANY();
<checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values. The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
FOLLOWPUSH(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)>ctx-><scope:delegateName()>-><endif><rule.name>(ctx<if(scope)>-
><scope:delegateName()><endif><if(args)>,
 <args; separator=", "><endif>);<\n>
FOLLOWPOP();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
/* <description> */
<if(label)>
{
   ANTLR3_MARKER <label>Start<elementIndex> = GETCHARINDEX();
   <if(scope)>ctx-><scope:delegateName()>-><endif>m<rule.name>(ctx<if(scope)>-
><scope:delegateName()><endif> <if(args)>, <endif><args; separator=", ">);
   <checkRuleBacktrackFailure()>
   <label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
   <label>->setType(<label>, ANTLR3_TOKEN_INVALID);
```

```
  <label>->setStartIndex(<label>, <label>Start<elementIndex>);
  <label>->setStopIndex(<label>, GETCHARINDEX()-1);

  <label>->input = INPUT;
}
<else>
<if(scope)>ctx-><scope:delegateName()>-><endif>m<rule.name>(ctx<if(scope)>-
><scope:delegateName()><endif> <if(args)>, <endif><args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
{
  ANTLR3_UINT32 <label>Start<elementIndex>;
  <labelType> <label>;
  <label>Start<elementIndex> = GETCHARINDEX();
  MATCHC(ANTLR3_CHARSTREAM_EOF);
  <checkRuleBacktrackFailure()>
  <label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
  <label>->setType(<label>, ANTLR3_TOKEN_EOF);
  <label>->setStartIndex(<label>, <label>Start<elementIndex>);
  <label>->setStopIndex(<label>, GETCHARINDEX()-1);
  <label>->input = INPUT->tnstream->istream;
}
<else>
  MATCHC(ANTLR3_CHARSTREAM_EOF);
  <checkRuleBacktrackFailure()>
   <endif>
>>

// used for left-recursive rules
recRuleDefArg()                       ::= "int <recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<
```

```
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( LA(1)==ANTLR3_TOKEN_DOWN ) {
   MATCHT(ANTLR3_TOKEN_DOWN, NULL);
   <checkRuleBacktrackFailure()>
   <children:element()>
   MATCHT(ANTLR3_TOKEN_UP, NULL);
   <checkRuleBacktrackFailure()>
}
<else>
MATCHT(ANTLR3_TOKEN_DOWN, NULL);
<checkRuleBacktrackFailure()>
<children:element()>
MATCHT(ANTLR3_TOKEN_UP, NULL);
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate
 (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) )
{
   <ruleBacktrackFailure()>
   <newFPE(...)>
}
>>

newFPE() ::= <<
   CONSTRUCTEX();
   EXCEPTION->type         = ANTLR3_FAILED_PREDICATE_EXCEPTION;
   EXCEPTION->message      = (void *)"<description>";
   EXCEPTION->ruleName  = (void *)"<ruleName>";
   <\n>
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

{
   int LA<decisionNumber>_<stateNumber> = LA(<k>);
   <edges; separator="\nelse ">
   else
   {
```

```
<if(eotPredictsAlt)>
      alt<decisionNumber>=<eotPredictsAlt>;
<else>
      <ruleBacktrackFailure()>

      <newNVException()>
      goto rule<ruleDescriptor.name>Ex;

<endif>
   }
}
>>

newNVException() ::= <<
CONSTRUCTEX();
EXCEPTION->type         = ANTLR3_NO_VIABLE_ALT_EXCEPTION;
EXCEPTION->message      = (void *)"<description>";
EXCEPTION->decisionNum
  = <decisionNumber>;
EXCEPTION->state        = <stateNumber>;
<@noViableAltException()>
<\n>
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
   int LA<decisionNumber>_<stateNumber> = LA(<k>);
   <edges; separator="\nelse ">
}
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/

dfaLoopbackStateDecls()::= <<
ANTLR3_UINT32   LA<decisionNumber>_<stateNumber>;
>>
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
```

```
 ::= <<
{
 /* dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
  */
  int LA<decisionNumber>_<stateNumber> = LA(<k>);
  <edges; separator="\nelse "><\n>
  <if(eotPredictsAlt)>
  <if(!edges)>
 alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
 <else>
  else
  {
 alt<decisionNumber>=<eotPredictsAlt>;
  }<\n>
  <endif>
  <endif>
}
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( <if(predicates)>(<predicates>) && <endif>(<labelExpr>))
{
  <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.
 The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k>) )
{
<edges; separator="\n">

default:
<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>;
<else>
```

```
   <ruleBacktrackFailure()>
   <newNVException()>
   goto rule<ruleDescriptor.name>Ex;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k>) )
{
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
   alt<decisionNumber>=<eotPredictsAlt>;
   break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it |case <it>:}; separator="\n">
 {
  <targetState>
 }
   break;
>>

// C y c l i c  D F A

/** The code to initiate execution of
 a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(ctx, RECOGNIZER, ISTREAM,
&cdfa<decisionNumber>);
<checkRuleBacktrackFailure()>
>>

/* Dump DFA tables as static initialized arrays of shorts(16 bits)/characters(8 bits)
```

```
* which are then used to statically initialize the dfa structure, which means that there
* is no runtime initialization whatsoever, other than anything the C compiler might
* need to generate. In general the C compiler will lay out memory such that there is no
* runtime code required.
*/
cyclicDFA(dfa) ::= <<
/** Static dfa state tables for Cyclic dfa:
*    <dfa.description>
*/
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_eot[<dfa.numberOfStates>] =
   {
 <dfa.eot; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_eof[<dfa.numberOfStates>]
 =
   {
 <dfa.eof; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_min[<dfa.numberOfStates>] =
   {
 <dfa.min; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_max[<dfa.numberOfStates>] =
   {
 <dfa.max; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_accept[<dfa.numberOfStates>] =
   {
 <dfa.accept; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_special[<dfa.numberOfStates>] =
   {
 <dfa.special; wrap="\n", separator=", ", null="-1">
   };

/** Used when there is no transition table entry for a particular state */
#define dfa<dfa.decisionNumber>_T_empty     NULL

<dfa.edgeTransitionClassMap.keys:{ table |
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_T<i0>[] =
   {
 <table; separator=", ", wrap="\n", null="-1">
   \};}; null = "">

/* Transition tables are
 a table of sub tables, with some tables
* reused for efficiency.
*/
```

```
static const ANTLR3_INT32 * const dfa<dfa.decisionNumber>_transitions[] =
{
   <dfa.transitionEdgeTables:{xref|dfa<dfa.decisionNumber>_T<xref>}; separator=", ", wrap="\n", null="_empty">
};

<if(dfa.specialStateSTs)>
static ANTLR3_INT32 dfa<dfa.decisionNumber>_sst(p<name> ctx, pANTLR3_BASE_RECOGNIZER recognizer,
pANTLR3_INT_STREAM is, pANTLR3_CYCLIC_DFA dfa, ANTLR3_INT32 s)
{
   ANTLR3_INT32    _s;

   _s     = s;
   switch  (s)
   {
   <dfa.specialStateSTs:{state |
   case <i0>:

 <state>}; separator="\n">
   }
<if(backtracking)>
   if (BACKTRACKING > 0)
   {
 FAILEDFLAG = ANTLR3_TRUE;
 return -1;
   }
<endif>

   CONSTRUCTEX();
   EXCEPTION->type         = ANTLR3_NO_VIABLE_ALT_EXCEPTION;
   EXCEPTION->message      = (void *)"<dfa.description>";
   EXCEPTION->decisionNum  = <dfa.decisionNumber>;
   EXCEPTION->state        = _s;
   <@noViableAltException()>
   return -1;
}
<endif>

<@errorMethod()>

/*
 Declare tracking structure for Cyclic DFA <dfa.decisionNumber>
*/
static
ANTLR3_CYCLIC_DFA cdfa<dfa.decisionNumber>
   = {
     <dfa.decisionNumber>,      /* Decision number of this dfa     */
     /* Which decision this represents:   */
     (const pANTLR3_UCHAR)"<dfa.description>",
<if(dfa.specialStateSTs)>
     (CDFA_SPECIAL_FUNC) dfa<dfa.decisionNumber>_sst,
<else>
     (CDFA_SPECIAL_FUNC) antlr3dfaspecialStateTransition, /* Default special state transition function */
<endif>

     antlr3dfaspecialTransition,  /* DFA specialTransition is currently just a default function in the runtime */
     antlr3dfapredict,   /* DFA simulator function is in the runtime */
     dfa<dfa.decisionNumber>_eot,     /* EOT table       */
     dfa<dfa.decisionNumber>_eof,     /* EOF table       */
     dfa<dfa.decisionNumber>_min,     /* Minimum tokens for each state    */
     dfa<dfa.decisionNumber>_max,     /* Maximum tokens for each state    */
     dfa<dfa.decisionNumber>_accept, /* Accept
 table       */
     dfa<dfa.decisionNumber>_special, /* Special transition states     */
     dfa<dfa.decisionNumber>_transitions /* Table of transition tables     */

 };
/* End of Cyclic DFA <dfa.decisionNumber>
* ---------------------
*/
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
   ANTLR3_UINT32 LA<decisionNumber>_<stateNumber>;<\n>
   ANTLR3_MARKER index<decisionNumber>_<stateNumber>;<\n>

 LA<decisionNumber>_<stateNumber> = LA(1);<\n>
   <if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
   index<decisionNumber>_<stateNumber> = INDEX();<\n>
   REWINDLAST();<\n>
   <endif>
   s = -1;
   <edges; separator="\nelse ">
 <if(semPredState)> <! return input cursor to state before we rewound !>
 SEEK(index<decisionNumber>_<stateNumber>);<\n>
 <endif>
   if ( s>=0 )
   {
 return s;
   }
}
break;
>>
```

```
/** Just like
 a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( <if(predicates)>(<predicates>) && <endif>(<labelExpr>) )
{
   s = <targetStateNumber>;
}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "( (<left>) && (<right>) )"

orPredicates(operands) ::= "(<operands:{o|(<o>)}; separator=\"||\">)"

notPredicate(pred) ::= "!( <evalPredicate(pred,{})> )"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>(ctx)"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead
 directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
((LA<decisionNumber>_<stateNumber> >= <lower>) && (LA<decisionNumber>_<stateNumber> \<= <upper>))
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "((LA(<k>) >= <lower>) &&
(LA(<k>) \<= <upper>))"

setTest(ranges) ::= "<ranges; separator=\" || \">"
```

```
// A T T R I B U T E S

makeScopeSet() ::= <<
/* makeScopeSet()
*/
/** Definition of the <scope.name> scope variable tracking
*  structure. An instance of this structure is created by calling
*  <name>_<scope.name>Push().
*/
typedef struct  <scopeStruct(sname=scope.name,...)>_struct
{
   /** Function that the user may provide to be called when the
    *  scope is destroyed (so you can free pANTLR3_HASH_TABLES and so on)
    *
    * \param POinter to an instance of this typedef/struct
    */
   void    (ANTLR3_CDECL *free) (struct <scopeStruct(sname=scope.name,...)>_struct
 * frame);

   /* =============================================================================
    * Programmer defined variables...
    */
   <scope.attributes:{it |<it.decl>;}; separator="\n">

   /* End of programmer defined variables
    * =============================================================================
    */
}
   <scopeStruct(sname=scope.name,...)>, * <scopeType(sname=scope.name,...)>;

>>

globalAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

ruleAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>
```

```
globalAttributeScopeFuncDecl(scope) ::=
<<
/* globalAttributeScopeFuncDecl(scope)
*/
<if(scope.attributes)>
/* -----------------------------------------------------------------------------
* Function declaration for creating a <name>_<scope.name> scope set
*/
static <scopeType(sname=scope.name,...)>
   <scopePushName(sname=scope.name,...)>(p<name> ctx);
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope);
/* ----------------------------------------------------------------------------- */

<endif>
>>

globalAttributeScopeFuncMacro(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeFuncMacro(scope)
*/
/** Function  for popping the top value from a <scopeStack(sname=scope.name)>
*/
void
<scopePopName(sname=scope.name,...)>(p<name> ctx)
{
   // First see if the user defined a function they want to be called when a
   // scope is popped/freed.
   //
 // If the user supplied the scope entries with a free function,then call it first
 //
   if (SCOPE_TOP(<scope.name>)->free != NULL)
 {
       SCOPE_TOP(<scope.name>)->free(SCOPE_TOP(<scope.name>));
 }

   // Now we decrement the scope's upper limit bound. We do not actually pop the scope as
   // we want to reuse scope entries if we do continuous push and pops. Most scopes don't
    // next too far so we don't want to keep freeing and allocating them
   //
   ctx-><scopeStack(sname=scope.name,...)>_limit--;
   SCOPE_TOP(<scope.name>) = (<scopeType(sname=scope.name)>)(ctx-><scopeStack(sname=scope.name,...)>-
>get(ctx-><scopeStack(sname=scope.name,...)>, ctx-><scopeStack(sname=scope.name,...)>_limit - 1));
}
<endif>
>>

ruleAttributeScopeFuncDecl(scope) ::= <<
```

```
<if(scope.attributes)>
/* ruleAttributeScopeFuncDecl(scope)
*/
/* ----------------------------------------------------------------------------
* Function declarations for creating a <name>_<scope.name> scope set
*/
static <scopeType(sname=scope.name,...)>   <scopePushName(sname=scope.name,...)>(p<name> ctx);
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope);
/* ---------------------------------------------------------------------------- */

<endif>
>>

ruleAttributeScopeFuncMacro(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncMacro(scope)
*/
/** Function
 for popping the top value from a <scopeStack(sname=scope.name,...)>
*/
void
<scopePopName(sname=scope.name,...)>(p<name> ctx)
{
   // First see if the user defined a function they want to be called when a
   // scope is popped/freed.
   //
 // If the user supplied the scope entries with a free function,then call it first
 //
   if (SCOPE_TOP(<scope.name>)->free != NULL)
 {
      SCOPE_TOP(<scope.name>)->free(SCOPE_TOP(<scope.name>));
 }

   // Now we decrement the scope's upper limit bound. We do not actually pop the scope as
   // we want to reuse scope entries if we do continuous push and pops. Most scopes don't
   // next too far so we don't want to keep freeing and allocating them
   //
   ctx-><scopeStack(sname=scope.name,...)>_limit--;
   SCOPE_TOP(<scope.name>) = (<scopeType(sname=scope.name)>)(ctx-><scopeStack(sname=scope.name,...)>-
>get(ctx-><scopeStack(sname=scope.name,...)>, ctx-><scopeStack(sname=scope.name,...)>_limit - 1));
}

<endif>
>>

globalAttributeScopeDef(scope)
 ::=
<<
```

```
/* globalAttributeScopeDef(scope)
*/
<if(scope.attributes)>
/** Pointer to the  <scope.name> stack for use by <scopePushName(sname=scope.name)>()
*  and <scopePopName(sname=scope.name,...)>()
*/
pANTLR3_STACK <scopeStack(sname=scope.name)>;
ANTLR3_UINT32 <scopeStack(sname=scope.name)>_limit;
/** Pointer to the top of the stack for the global scope <scopeStack(sname=scope.name)>
*/
<scopeType(sname=scope.name,...)>    (*<scopePushName(sname=scope.name,...)>)(struct <name>_Ctx_struct *
ctx);
<scopeType(sname=scope.name,...)>    <scopeTopDecl(sname=scope.name,...)>;

<endif>
>>

ruleAttributeScopeDef(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDef(scope)
*/
/** Pointer to the  <scope.name> stack for use by <scopePushName(sname=scope.name)>()
*  and <scopePopName(sname=scope.name,...)>()
*/
pANTLR3_STACK <scopeStack(sname=scope.name,...)>;
ANTLR3_UINT32 <scopeStack(sname=scope.name,...)>_limit;
<scopeType(sname=scope.name,...)>   (*<scopePushName(sname=scope.name,...)>)(struct
 <name>_Ctx_struct * ctx);
<scopeType(sname=scope.name,...)>   <scopeTopDecl(sname=scope.name,...)>;

<endif>
>>

globalAttributeScopeFuncs(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeFuncs(scope)
*/
<attributeFuncs(scope)>
<endif>
>>

ruleAttributeScopeFuncs(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncs(scope)
*/
<attributeFuncs(scope)>
<endif>
>>
```

```
globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScope(scope)
*/
ctx-><scopePushName(sname=scope.name,...)>     = <scopePushName(sname=scope.name,...)>;
ctx-><scopeStack(sname=scope.name,...)>    = antlr3StackNew(0);
ctx-><scopeStack(sname=scope.name,...)>_limit    = 0;
<scopeTop(sname=scope.name,...)>      = NULL;
<endif>
>>

ruleAttributeScope(scope) ::=
<<
<if(scope.attributes)>
/* ruleAttributeScope(scope)
*/
ctx-><scopePushName(sname=scope.name,...)>     = <scopePushName(sname=scope.name,...)>;
ctx-><scopeStack(sname=scope.name,...)>    = antlr3StackNew(0);
ctx-><scopeStack(sname=scope.name,...)>_limit
    = 0;
<scopeTop(sname=scope.name,...)>      = NULL;
<endif>
>>
globalAttributeScopeFree(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScope(scope)
*/
ctx-><scopeStack(sname=scope.name,...)>->free(ctx-><scopeStack(sname=scope.name,...)>);
<endif>
>>

ruleAttributeScopeFree(scope) ::=
<<
<if(scope.attributes)>
/* ruleAttributeScope(scope)
*/
ctx-><scopeStack(sname=scope.name,...)>->free(ctx-><scopeStack(sname=scope.name,...)>);
<endif>
>>

scopeTopDecl(sname) ::= <<
p<name>_<sname>Top
>>

scopeTop(sname) ::= <<
ctx-><scopeTopDecl(sname=sname,...)>
>>
```

```
scopePop(sname) ::= <<
<scopePopName(sname=sname,...)>(ctx);
>>

scopePush(sname) ::= <<
p<name>_<sname>Push(ctx)
>>

scopePopName(sname) ::= <<
p<name>_<sname>Pop
>>

scopePushName(sname) ::= <<
p<name>_<sname>Push
>>

scopeType(sname) ::= <<
p<name>_<sname>_SCOPE
>>

scopeStruct(sname) ::= <<
<name>_<sname>_SCOPE
>>

scopeStack(sname) ::= <<
p<name>_<sname>Stack
>>

attributeFuncs(scope)
 ::= <<
<if(scope.attributes)>
/* attributeFuncs(scope)
*/

static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope)
{
   ANTLR3_FREE(scope);
}

/** \brief Allocate initial memory for a <name> <scope.name> scope variable stack entry and
*         add it to the top of the stack.
*
* \remark
* By default the structure is freed with ANTLR_FREE(), but you can use the
* the \@init action to install a pointer to a custom free() routine by
* adding the code:
* \code
*   <scopeTop(sname=scope.name)>->free = myroutine;
* \endcode
*
* With lots of comments of course! The routine should be declared in
* \@members { } as:
* \code
*   void ANTLR3_CDECL myfunc( <scopeType(sname=scope.name)> ptr).
* \endcode
*
* It should perform any custom freeing stuff that you need (call ANTLR_FREE3, not free()
* NB: It should not free the pointer it is given, which is the scope stack entry itself
* and will be freed by the function that calls your custom
 free routine.
*
*/
static <scopeType(sname=scope.name)>
<scopePushName(sname=scope.name)>(p<name> ctx)
{
   /* Pointer used to create a new set of attributes
    */
   <scopeType(sname=scope.name)>      newAttributes;

   /* Allocate the memory for a new structure if we need one.
    */
   if (ctx-><scopeStack(sname=scope.name)>->size(ctx-><scopeStack(sname=scope.name)>) > ctx-
><scopeStack(sname=scope.name)>_limit)
   {
       // The current limit value was less than the number of scopes available on the stack so
       // we can just reuse one. Our limit tracks the stack count, so the index of the entry we want
       // is one less than that, or conveniently, the current value of limit.
       //
       newAttributes = (<scopeType(sname=scope.name)>)ctx-><scopeStack(sname=scope.name)>->get(ctx-
><scopeStack(sname=scope.name)>, ctx-><scopeStack(sname=scope.name)>_limit);
   }
   else
   {
       // Need a new allocation
       //
       newAttributes = (<scopeType(sname=scope.name)>)
 ANTLR3_MALLOC(sizeof(<scopeStruct(sname=scope.name)>));
       if  (newAttributes != NULL)
       {
           /* Standard ANTLR3 library implementation
            */
           ctx-><scopeStack(sname=scope.name)>->push(ctx-><scopeStack(sname=scope.name)>, newAttributes,
(void (*)(void *))<scope.name>Free);
       }
   }
   // Blank out any previous free pointer, the user might or might install a new one.
   //
   newAttributes->free = NULL;

   // Indicate the position in the available stack that the current level is at
   //
   ctx-><scopeStack(sname=scope.name)>_limit++;

 /* Return value is the pointer to the new entry, which may be used locally
  * without de-referencing via the context.
    */
   return  newAttributes;
}<\n>

<endif>
>>
returnStructName(r) ::= "<r.name>_return"

returnType() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
ANTLR3_BOOLEAN
<endif>
%>

/**
 Generate the C type associated with a single or multiple return
*  value(s).
*/
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>_<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
```

```
<endif>
%>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "0".
*/
initValue(typeName) ::= <<
= <cTypeInitMap.(typeName)>
>>

/** Define a rule label  */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text>;
#undef RETURN_TYPE_<label.label.text>
#define RETURN_TYPE_<label.label.text> <ruleLabelType(referencedRule=label.referencedRule)><\n>
>>
/**
  Rule label default value */
ruleLabelInitVal(label) ::= <<
>>

ASTLabelType() ::=
"<if(recognizer.ASTLabelType)><recognizer.ASTLabelType><else>pANTLR3_BASE_TREE<endif>"

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
typedef struct <ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>_struct
{
<if(!TREE_PARSER)>
   /** Generic return elements for ANTLR3 rules that are not in tree parsers or returning trees
    */
   pANTLR3_COMMON_TOKEN    start;
   pANTLR3_COMMON_TOKEN    stop;
<else>
   <recognizer.ASTLabelType>       start;
   <recognizer.ASTLabelType>       stop;
<endif>
   <@ruleReturnMembers()>
   <ruleDescriptor.returnScope.attributes:{it |<it.type> <it.name>;}; separator="\n">
}
   <ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>;<\n><\n>
<endif>
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name>=<expr>;"

/** Note that the scopeAttributeRef does not have access to the
* grammar name directly
*/
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get( ctx->SCOPE_STACK(<scope>), ctx-
>SCOPE_STACK(<scope>)->size(ctx->SCOPE_STACK(<scope>)) - <negIndex> - 1) ))-><attr.name>
<else>
<if(index)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get(ctx->SCOPE_STACK(<scope>),
(ANTLR3_UINT32)<index> ) ))-><attr.name>
<else>
 (SCOPE_TOP(<scope>))-><attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get( ctx->SCOPE_STACK(<scope>),
 ctx->SCOPE_STACK(<scope>)->size(ctx->SCOPE_STACK(<scope>)) - <negIndex> - 1) ))-><attr.name> =
<expr>;
<else>
<if(index)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get(ctx->SCOPE_STACK(<scope>),
(ANTLR3_UINT32)<index> ) ))-><attr.name> = <expr>;
<else>
 (SCOPE_TOP(<scope>))-><attr.name>=<expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
```

```
*/
isolatedDynamicScopeRef(scope) ::= "ctx->SCOPE_STACK(<scope>)"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<scope>.<attr.name>
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr)
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>=<expr>;
<else>
<attr.name>=<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach
//
tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>->getText(<scope>))"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>->getType(<scope>))"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>->getLine(<scope>))"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>->getCharPositionInLine(<scope>))"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>->getChannel(<scope>))"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>->getTokenIndex(<scope>))"
tokenLabelPropertyRef_tree(scope,attr) ::= "(<scope>->tree)"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>->getText(<scope>)->toInt32(<scope>->getText(<scope>)))"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>.start)"
```

```
ruleLabelPropertyRef_stop(scope,attr)
 ::= "(<scope>.stop)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>.tree)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(STRSTREAM->toStringSS(STRSTREAM, <scope>.start, <scope>.start))
<else>
(STRSTREAM->toStringTT(STRSTREAM, <scope>.start, <scope>.stop))
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(<scope>->getType(<scope>))"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(<scope>->getLine(<scope>))"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(<scope>->getCharPositionInLine(<scope>))"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(<scope>->getChannel(<scope>))"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(<scope>->getTokenIndex(<scope>))"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(<scope>->getText(<scope>))"

// Somebody may ref $template
 or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop"
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
INPUT->toStringSS(INPUT, ADAPTOR->getTokenStartIndex(ADAPTOR, retval.start), ADAPTOR-
>getTokenStopIndex(ADAPTOR, retval.start))
<else>
STRSTREAM->toStringTT(STRSTREAM, retval.start, LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "LEXER->getText(LEXER)"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "LEXSTATE->tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "LEXSTATE->tokenStartCharPositionInLine"
lexerRulePropertyRef_channel(scope,attr) ::= "LEXSTATE->channel"
lexerRulePropertyRef_start(scope,attr) ::= "LEXSTATE->tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(LEXER->getCharIndex(LEXER)-1)"
lexerRulePropertyRef_index(scope,attr)
 ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_int(scope,attr) ::= "LEXER->getText(LEXER)->toInt32(LEXER->getText(LEXER))"
```

```
// setting $st and $tree is allowed in local rule. everything else is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree=<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st=<expr>;"


/** How to deal with an @after for C targets. Because we cannot rely on
*  any garbage collection, after code is executed even in backtracking
*  mode. Must be documented clearly.
*/
execAfter(action) ::= <<
{
   <action>
}
>>

/** How to execute an action (when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> )
{
   <action>
}
<else>
if ( BACKTRACKING == 0 )
{
   <action>
}
<endif>
<else>
{
   <action>
}
<endif>
>>

// M I S C (properties, etc...)

bitsetDeclare(name, words64) ::= <<

/** Bitset defining follow
 set for error recovery in rule state: <name>  */
static ANTLR3_BITWORD <name>_bits[] = { <words64:{it |ANTLR3_UINT64_LIT(<it>)}; separator=", "> };
static  ANTLR3_BITSET_LIST <name> = { <name>_bits, <length(words64)> };
>>
```

```
bitset(name, words64) ::= <<
antlr3BitsetSetAPI(&<name>);<\n>
>>

codeFileExtension() ::= ".c"

true_value() ::= "ANTLR3_TRUE"
false_value() ::= "ANTLR3_FALSE"
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/C.stg

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2005-2008 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2008-2009 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
 OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template overrides to add debugging to normal Java output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
```

```
@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.Runtime.Debug;
using IOException = System.IO.IOException;
>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
public static readonly
 string[] ruleNames =
 new string[]
 {
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n ", separator=", ">
 };<\n>
<endif>
<if(grammar.grammarIsRoot)><! grammar imports other grammar(s) !>
int ruleLevel = 0;
public virtual int RuleLevel { get { return ruleLevel; } }
public virtual void IncRuleLevel() { ruleLevel++; }
public virtual void DecRuleLevel() { ruleLevel--; }
<if(profile)>
<ctorForProfilingRootGrammar()>
<else>
<ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else><! imported grammar !>
public int RuleLevel { get { return <grammar.delegators:{g| <g:delegateName()>}>.RuleLevel; } }
public void IncRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.IncRuleLevel(); }
public void DecRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.DecRuleLevel(); }
<ctorForDelegateGrammar()>
<endif>
<if(profile)>
public override bool AlreadyParsedRule( IIntStream input, int ruleIndex )
{
 int stopIndex = GetRuleMemoization(ruleIndex, input.Index);
 ((Profiler)dbg).ExamineRuleMemoization(input,
 ruleIndex, stopIndex, <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
 return base.AlreadyParsedRule(input, ruleIndex);
}<\n>
public override void Memoize( IIntStream input, int ruleIndex, int ruleStartIndex )
{
 ((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
 base.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
```

```
protected virtual bool EvalPredicate( bool result, string predicate )
{
 dbg.SemanticPredicate( result, predicate );
 return result;
}<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
 : this( input, DebugEventSocketProxy.DefaultDebuggerPort, new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier;
 null="public"> <name>( <inputStreamType> input, int port, RecognizerSharedState state )
 : base( input, state )
{
 <createListenerAndHandshake()>
 <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}> );}; separator="\n">
 <parserCtorBody()>
 <@finally()>
}<\n>
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
 : this( input, new Profiler(null), new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state )
 : base( input, dbg, state )
{
 Profiler p = (Profiler)dbg;
 p.setParser(this);
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state, this<grammar.delegators:{g|,
<g:delegateName()>}> );}; separator="\n">
 <parserCtorBody()>
 <@finally()>
}
<\n>
>>

/**
```

```
 Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}> )
 : base( input, dbg, state )
{
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}> );}; separator="\n">
 <parserCtorBody()>
}<\n>
>>

ctorForPredefinedListener() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg
)
 <@superClassRef>: base( input, dbg, new RecognizerSharedState() )<@end>
{
<if(profile)>
 Profiler p = (Profiler)dbg;
 p.setParser(this);
<endif>
 <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|,
 <g:delegateName()>}>);}; separator="\n">
 <parserCtorBody()>
 <@finally()>
}<\n>
>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, input.TreeAdaptor );<\n>
<else>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, null );<\n>
<endif>
DebugListener = proxy;
try
{
 proxy.Handshake();
}
catch ( IOException ioe )
{
 ReportError( ioe );
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"
```

```
/*
* Much of the following rules were merged into CSharp3.stg.
*/

@rule.preamble() ::= <<
if (RuleLevel == 0)
 DebugListener.Commence();
IncRuleLevel();
>>
//@rule.preamble() ::= <<
//try
//{
// dbg.EnterRule( GrammarFileName, "<ruleName>" );
// if ( RuleLevel == 0 )
// {
//  dbg.Commence();
// }
// IncRuleLevel();
// dbg.Location( <ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine> );<\n>
//>>

@rule.postamble() ::= <<
DecRuleLevel();
if (RuleLevel == 0)
 DebugListener.Terminate();
>>
//@rule.postamble()
 ::= <<
//dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
//}
//finally
//{
// dbg.ExitRule( GrammarFileName, "<ruleName>" );
// DecRuleLevel();
// if ( RuleLevel == 0 )
// {
//  dbg.Terminate();
// }
//}<\n>
//>>

//@insertSynpreds.start() ::= "dbg.BeginBacktrack( state.backtracking );"
//@insertSynpreds.stop() ::= "dbg.EndBacktrack( state.backtracking, success );"

// Common debug event triggers used by region overrides below

//enterSubRule() ::= <<
```

```
//try
//{
// dbg.EnterSubRule( <decisionNumber> );<\n>
//>>

//exitSubRule() ::= <<
//}
//finally
//{
// dbg.ExitSubRule( <decisionNumber> );
//}<\n>
//>>

//enterDecision() ::= <<
//try
//{
// dbg.EnterDecision( <decisionNumber> );<\n>
//>>

//exitDecision() ::= <<
//}
//finally
//{
// dbg.ExitDecision( <decisionNumber> );
//}<\n>
//>>

//enterAlt(n) ::= "dbg.EnterAlt( <n> );<\n>"

// Region overrides that tell various constructs to add debugging
 triggers

//@block.predecision() ::= "<enterSubRule()><enterDecision()>"

//@block.postdecision() ::= "<exitDecision()>"

//@block.postbranch() ::= "<exitSubRule()>"

//@ruleBlock.predecision() ::= "<enterDecision()>"

//@ruleBlock.postdecision() ::= "<exitDecision()>"

//@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

//@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

//@positiveClosureBlock.preloop() ::= "<enterSubRule()>"
```

```
//@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

//@positiveClosureBlock.predecision() ::= "<enterDecision()>"

//@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

//@positiveClosureBlock.earlyExitException() ::=
// "dbg.RecognitionException( eee<decisionNumber> );<\n>"

//@closureBlock.preloop() ::= "<enterSubRule()>"

//@closureBlock.postloop() ::= "<exitSubRule()>"

//@closureBlock.predecision() ::= "<enterDecision()>"

//@closureBlock.postdecision() ::= "<exitDecision()>"

//@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

//@element.prematch()
 ::=
// "dbg.Location( <it.line>, <it.pos> );"

//@matchSet.mismatchedSetException() ::=
// "dbg.RecognitionException( mse );"

//@dfaState.noViableAltException() ::= "dbg.RecognitionException( nvae );"

//@dfaStateSwitch.noViableAltException() ::= "dbg.RecognitionException( nvae );"

//dfaDecision(decisionNumber,description) ::= <<
//try
//{
// isCyclicDecision = true;
// <super.dfaDecision(...)>
//}
//catch ( NoViableAltException nvae )
//{
// dbg.RecognitionException( nvae );
// throw nvae;
//}
//>>

//@cyclicDFA.errorMethod() ::= <<
//public override void Error( NoViableAltException nvae )
//{
// ((DebugParser)recognizer).dbg.RecognitionException( nvae );
//}
```

```
//>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
EvalPredicate(<pred>, "<description>")
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/Dbg.stg
No license file was found, but licenses were detected in source scan.

```
/*
New style messages. This file contains the actual layout of the messages emitted by ANTLR.
The text itself is coming out of the languages/*stg files, according to the chosen locale.
This file contains the format that mimicks GCC output.
*/

group gnu;

location(file, line, column) ::= "<file>:<line>:"
```

```
message(id, text) ::= "<text> (<id>)"

report(location, message, type) ::= "<location> <type>: <message>"

wantsSingleLineMessage() ::= "true"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/messages/formats/gnu.stg
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Python by code generator.
*/
group ASTDbg;

astAccessor() ::= <<
def setTreeAdaptor(self, adaptor):
```

```
<if(grammar.grammarIsRoot)>
  self._adaptor = DebugTreeAdaptor(self.dbg, adaptor)
<else>
  self._adaptor = adaptor # delegator sends dbg adaptor
<endif>
  <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(self._adaptor)}>

def getTreeAdaptor(self):
   return self._adaptor

adaptor = property(getTreeAdaptor, setTreeAdaptor)<\n>
>>

createListenerAndHandshake() ::= <<
proxy = DebugEventSocketProxy(self,
adaptor=<if(TREE_PARSER)>self.input.getTreeAdaptor()<else>self._adaptor<endif>,
                          debug=debug_socket, port=port)
self.setDebugListener(proxy)
self.adaptor.setDebugListener(proxy)
self.input.setDebugListener(proxy)
#self.set<inputStreamType>(Debug<inputStreamType>(self.input, proxy))
proxy.handshake()
>>

@rewriteElement.pregen() ::= "self._dbg.location(<e.line>, <e.pos>)"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

```
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
<!protected TreeAdaptor adaptor = new CommonTreeAdaptor();<\n>!>
setTreeAdaptor: function(adaptor) {
  this.adaptor = adaptor;
  <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
},
getTreeAdaptor: function() {
  return this.adaptor;
},
>>

@returnScope.ruleReturnMembers()
 ::= <<
getTree: function() { return this.tree; }
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{var <it.label.text>_tree=null;}; separator="\n">
```

```
<ruleDescriptor.tokenListLabels:{var <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleTokenStream(this.adaptor,"token <it>");};
separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations()
 ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = this.adaptor.nil();<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite
```

```
 */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.getTree());
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.getTree());
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

//
 R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<
```

```
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;
<rewriteCodeLabels()>
root_0 = this.adaptor.nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree
 = this.adaptor.rulePostProcessing(root_0);
this.input.replaceChildren(this.adaptor.getParent(retval.start),
                  this.adaptor.getChildIndex(retval.start),
                  this.adaptor.getChildIndex(_last),
                  retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRule<rewriteElementType>Stream(this.adaptor,"token
<it>",<it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRule<rewriteElementType>Stream(this.adaptor,"token
<it>", list_<it>);};
   separator="\n"
```

```
>
<referencedWildcardLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"wildcard
 <it>",<it>);};
   separator="\n"
>
<referencedWildcardListLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"wildcard <it>",list_<it>);};
   separator="\n"
>
<referencedRuleLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"token
<it>",<it>!=null?<it>.tree:null);};
   separator="\n"
>
<referencedRuleListLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"token <it>",list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElementsDeep:{el
 | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
```

```
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext()}; separator="||">) ) {
   throw new org.antlr.runtime.tree.RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/**
 For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
var root_<treeLevel> = this.adaptor.nil();
<root:rewriteElement()>
<children:rewriteElement()>
this.adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
```

```
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
this.adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/**
 Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<createRewriteNodeFromElement(...)>, root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this
```

```
 rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
this.adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>,
 stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>
```

```
createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
this.adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.nextToken()<if(args)>,
 <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
this.adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/JavaScript/AST.stg

No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2006 Kay Roepke 2010 Alan Condit
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

```
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/
parserMembers() ::= <<
DebugTreeAdaptor *adaptor = [DebugTreeAdaptor newTreeAdaptor:(id)dbg Adaptor:[CommonTreeAdaptor
newTreeAdaptor]];

// fix this
- (void) setTreeAdaptor:(id<TreeAdaptor>)anAdaptor
{
   adaptor = [DebugTreeAdaptor newTreeAdaptor:dbg Adaptor:anAdaptor];
<if(grammar.grammarIsRoot)>
   adaptor = [DebugTreeAdaptor
 newTreeAdaptor:adaptor withDBG:dbg];
<else>
   adaptor = (DebugTreeAdaptor *)adaptor; // delegator sends dbg adaptor
<endif><\n>
   <grammar.directDelegates:{g|[<g:delegateName()> setTreeAdaptor:adaptor];}>
}

- (id<TreeAdaptor>)getTreeAdaptor
{
   return adaptor;
}<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy =
   [DebugEventSocketProxy newDebugEventSocketProxy:self, port, <if(TREE_PARSER)>[input
getTreeAdaptor]<else>adaptor<endif>];
[self setDebugListener:proxy];
[self set<inputStreamType>:[Debug<inputStreamType> newDebug<inputStreamType>:input with:proxy]];
try {
   [proxy handshake];
}
@catch (IOException *ioe) {
  [self reportError:ioe];
}
>>

@ctorForRootGrammar.finally() ::= <<
CommonTreeAdaptor *adap = [CommonTreeAdaptor newTreeAdaptor];
[self setTreeAdaptor:adap];
[proxy setTreeAdaptor:adap];
>>

@ctorForProfilingRootGrammar.finally() ::=<<
CommonTreeAdaptor *adap = [CommonTreeAdaptor
 newTreeAdaptor];
[self setTreeAdaptor:adap];
[proxy setTreeAdaptor:adap];
>>

@ctorForPredefinedListener.superClassRef() ::= @"super(input, dbg);"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>
CommonTreeAdaptor *adap = [CommonTreeAdaptor newTreeAdaptor];
[self setTreeAdaptor:adap];<\n>
<endif>
>>

@treeParserHeaderFile.superClassName ::= "DebugTreeParser"

@rewriteElement.pregen() ::= "[debugListener locationLine:<e.line> column:<e.pos>];"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

```
*/

/** The API version of the runtime that recognizers generated by this runtime
*  need.
*/
apiVersion() ::= "1"

// System.Boolean.ToString() returns "True" and "False", but the proper C# literals are "true" and "false"
// The Java version of Boolean returns "true" and "false", so they map to themselves here.
booleanLiteral ::= [
	"True":"true",
	"False":"false",
	"true":"true",
	"false":"false",
	default:"false"
]

/** The overall file structure
 of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
           bitsets, buildTemplate, buildAST, rewriteMode, profile,
           backtracking, synpreds, memoize, numRules,
           fileName, ANTLRVersion, generatedTimestamp, trace,
           scopes, superClass, literals) ::=
<<
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
import sys
from antlr3 import *
```

```
<if(TREE_PARSER)>
from antlr3.tree import *<\n>
<endif>
from antlr3.compat import set, frozenset
<@end>

<actions.(actionScope).header>

<! <docComment> !>

# for convenience in actions
HIDDEN = BaseRecognizer.HIDDEN

# token types
<tokens:{it | <it.name>=<it.type>}; separator="\n">

<recognizer>

<if(actions.(actionScope).main)>
<actions.(actionScope).main>
<else>
def main(argv, stdin=sys.stdin, stdout=sys.stdout, stderr=sys.stderr):
<if(LEXER)>
     from antlr3.main import LexerMain
    main = LexerMain(<recognizer.name>)<\n>
<endif>
<if(PARSER)>
    from antlr3.main import ParserMain
    main = ParserMain("<recognizer.grammar.name>Lexer", <recognizer.name>)<\n>
<endif>
<if(TREE_PARSER)>
    from antlr3.main import WalkerMain
    main = WalkerMain(<recognizer.name>)<\n>
<endif>
    main.stdin = stdin
    main.stdout = stdout
    main.stderr = stderr
    main.execute(argv)<\n>
<endif>

<actions.(actionScope).footer>

if __name__ == '__main__':
    main(sys.argv)

>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode,
    labelType="CommonToken", superClass="Lexer") ::= <<
<grammar.directDelegates:
{g|from <g.recognizerName> import <g.recognizerName>}; separator="\n">

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
   <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>

   grammarFileName = "<fileName>"
   api_version = <apiVersion()>

    def __init__(self<grammar.delegators:{g|, <g:delegateName()>}>, input=None, state=None):
       if state is None:
           state = RecognizerSharedState()
       super(<grammar.recognizerName>, self).__init__(input, state)

<if(memoize)>
<if(grammar.grammarIsRoot)>
       self._state.ruleMemo = {}
<endif>
<endif>

       <grammar.directDelegates:
        {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)}; separator="\n">
       <grammar.directDelegates:
        {g|<g.delegates:{h|self.<h:delegateName()> = self.<g:delegateName()>.<h:delegateName()>};
separator="\n">}; separator="\n">
       <grammar.delegators:
        {g|self.<g:delegateName()> = <g:delegateName()>}; separator="\n">
       <last(grammar.delegators):
     {g|self.gParent = <g:delegateName()>}; separator="\n">
       self.delegates = [<grammar.delegates: {g|self.<g:delegateName()>}; separator = ", ">]

       <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>};
 separator="\n">

       <actions.lexer.init>



   <actions.lexer.members>



<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>
   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>


>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
def nextToken(self):
   while True:
       if self.input.LA(1) == EOF:
           return self.makeEOFToken()

       self._state.token = None
       self._state.channel = DEFAULT_CHANNEL
       self._state.tokenStartCharIndex = self.input.index()
       self._state.tokenStartCharPositionInLine = self.input.charPositionInLine
       self._state.tokenStartLine
 = self.input.line
       self._state._text = None
       try:
           m = self.input.mark()
           try:
               # means we won't throw slow exception
               self._state.backtracking = 1
               try:
                   self.mTokens()
               finally:
                   self._state.backtracking = 0

           except BacktrackingFailed:
               # mTokens backtracks with synpred at backtracking==2
               # and we set the synpredgate to allow actions at level 1.
               self.input.rewind(m)
               self.input.consume() # advance one char and try again

           else:
               self.emit()
               return self._state.token

       except RecognitionException, re:
           # shouldn't happen in backtracking mode, but...
           self.reportError(re)
```

```
        self.recover(re)



def memoize(self, input, ruleIndex, ruleStartIndex, success):
   if self._state.backtracking > 1:
       #
 is Lexer always superclass?
       super(<grammar.recognizerName>, self).memoize(input, ruleIndex, ruleStartIndex, success)



def alreadyParsedRule(self, input, ruleIndex):
   if self._state.backtracking > 1:
       return super(<grammar.recognizerName>, self).alreadyParsedRule(input, ruleIndex)
   return False



>>

actionGate() ::= "self._state.backtracking == 0"

filteringActionGate() ::= "self._state.backtracking == 1"

/** How to generate a parser */

genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass, labelType, members,
       rewriteElementType, filterMode, init, ASTLabelType="Object") ::= <<
<if(grammar.grammarIsRoot)>
# token names
tokenNames = [
   "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>",
   <tokenNames; wrap, separator=", ">
]<\n>
<else>
from <grammar.composite.rootGrammar.recognizerName> import tokenNames<\n>
<endif>
<scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScopeClass(scope=it)><endif>}>

<grammar.directDelegates:
 {g|from <g.recognizerName> import <g.recognizerName>}; separator="\n">

<rules:{it|<ruleAttributeScopeClass(scope=it.ruleDescriptor.ruleScope)>}>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
   grammarFileName = "<fileName>"
   api_version = <apiVersion()>
   tokenNames = tokenNames
```

```
  def __init__(self<grammar.delegators:{g|, <g:delegateName()>}>, input, state=None, *args, **kwargs):
    if state is None:
      state = RecognizerSharedState()

    <@args()>
    super(<grammar.recognizerName>, self).__init__(input, state, *args, **kwargs)

<if(memoize)>
<if(grammar.grammarIsRoot)>
    self._state.ruleMemo = {}
<endif>
<endif>

    <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">

    <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeStack(scope=it)><endif>}>
 <rules:{it | <ruleAttributeScopeStack(scope=it.ruleDescriptor.ruleScope)>}>

    <init>

    <grammar.delegators:
     {g|self.<g:delegateName()> = <g:delegateName()>}; separator="\n">
    <grammar.directDelegates:
     {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)}; separator="\n">
    <grammar.directDelegates:
     {g|<g.delegates:{h|self.<h:delegateName()> = self.<g:delegateName()>.<h:delegateName()>};
separator="\n">}; separator="\n">
    <last(grammar.delegators):
   {g|self.gParent = self.<g:delegateName()>}; separator="\n">
    self.delegates = [<grammar.delegates: {g|self.<g:delegateName()>}; separator = ", ">]

 <@init><@end>


  <@members><@end>

  <members>

  <rules; separator="\n\n">

  <! generate rule/method definitions for imported rules so they
     appear to be defined in this recognizer. !>
  <grammar.delegatedRules:{ruleDescriptor| <delegateRule(ruleDescriptor)> }; separator="\n">

  <synpreds:{p | <synpred(p)>}>

  <cyclicDFAs:cyclicDFA()> <! dump
 tables for all DFA !>

   <bitsets:{it | FOLLOW_<it.name>_in_<it.inName><it.tokenIndex> = frozenset([<it.tokenTypes:{it |
<it>};separator=", ">])<\n>}>

>>

delegateRule(ruleDescriptor) ::= <<
def <ruleDescriptor.name>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<\ >   <if(ruleDescriptor.hasReturnValue)>return
<endif>self.<ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<if(ruleDescriptor.parameterScope)
><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", "><endif>)



>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
      ASTLabelType="Object", superClass="Parser", labelType="Token",
      members={<actions.parser.members>},
      init={<actions.parser.init>}
      ) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
             bitsets, "TokenStream", superClass,
             labelType, members, "Token",
             false, init, ASTLabelType)>
>>

/** How to generate a tree
 parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
          numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
 members={<actions.treeparser.members>},
    init={<actions.treeparser.init>}
          ) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
             bitsets, "TreeNodeStream", superClass,
             labelType, members, "Node",
             filterMode, init, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates
```

```
 cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start "<ruleName>"
def <ruleName>_fragment(self, <ruleDescriptor.parameterScope:parameterScope()>):
    <ruleLabelDefs()>
<if(trace)>
    self.traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
    try:
        <block>

    finally:
        self.traceOut("<ruleName>_fragment", <ruleDescriptor.index>)

<else>
    <block>
<endif>
# $ANTLR end "<ruleName>"



>>


synpred(name) ::= <<
def <name>(self):
    self._state.backtracking += 1
    <@start()>
    start = self.input.mark()
    try:
        self.<name>_fragment()
    except BacktrackingFailed:
        success = False
    else:
        success = True
    self.input.rewind(start)
    <@stop()>
    self._state.backtracking -= 1
    return success



>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
```

```
<if(memoize)>
if self._state.backtracking
 > 0 and self.alreadyParsedRule(self.input, <ruleDescriptor.index>):
   # for cached failed rules, alreadyParsedRule will raise an exception
   success = True
   return <ruleReturnValue()>

<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if self._state.backtracking > 0:
   raise BacktrackingFailed

<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<returnScope(scope=ruleDescriptor.returnScope)>

# $ANTLR start "<ruleName>"
# <fileName>:<description>
<ruleDescriptor.actions.decorate>
def <ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<if(trace)>
   self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
    <@preamble()>
   <@body><ruleBody()><@end>
   <@postamble()>
   return <ruleReturnValue()>

# $ANTLR end "<ruleName>"
>>

ruleBody() ::= <<
<if(memoize)>
<if(backtracking)>
success = False<\n>
```

```
<endif>
<endif>
try:
   try:
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>

<if(memoize)>
<if(backtracking)>
       success = True<\n>
<endif>
<endif>
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   except RecognitionException, re:
       self.reportError(re)
       self.recover(self.input, re)
       <@setErrorReturnValue()>

<endif>
<else>
   finally:
       pass

<endif>
<endif>
finally:
<if(trace)>
   self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
   <memoize()>
   <ruleScopeCleanUp()>
   <finally>
   pass
>>

catch(decl,action)
 ::= <<
except <e.decl>:
   <e.action>
```

```
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval = self.<ruleDescriptor.name>_return()
retval.start = self.input.LT(1)<\n>
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name> = <if(a.initValue)><a.initValue><else>None<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex = self.input.index()
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it | self.<it>_stack.append(<it>_scope())}; separator="\n">
<ruleDescriptor.ruleScope:{it | self.<it.name>_stack.append(<it.name>_scope())}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it | self.<it>_stack.pop()}; separator="\n">
<ruleDescriptor.ruleScope:{it | self.<it.name>_stack.pop()}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
    :{it | <it.label.text> = None}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,
 ruleDescriptor.wildcardTreeListLabels]
   :{it | list_<it.label.text> = None}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{it | <it.label.text> = None}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it | <it.label.text> = None}; separator="\n"
>
<ruleDescriptor.charLabels:{it | <it.label.text> = None}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
```

```
   :{it | list_<it.label.text> = None}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop
 = self.input.LT(-1)<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if self._state.backtracking > 0:
   self.memoize(self.input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex, success)

<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start "<ruleName>"
def m<ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<if(trace)>
   self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
<if(memoize)>
<if(backtracking)>
```

```
   success = False<\n>
<endif>
<endif>
   try:
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       _type = <ruleName>
       _channel = DEFAULT_CHANNEL

       <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       self._state.type = _type
       self._state.channel = _channel
       <(ruleDescriptor.actions.after):execAction()>
<endif>
<if(memoize)>
<if(backtracking)>
       success = True<\n>
<endif>
<endif>

   finally:
<if(trace)>
       self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
 <ruleScopeCleanUp()>
       <memoize()>
       pass

# $ANTLR end "<ruleName>"



>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
def mTokens(self):
   <block><\n>
```

```
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@body><blockBody()><@end>
>>

blockBody()
 ::= <<
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<@prebranch()>
<alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
```

```
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n)
 ::= <<
# <fileName>:<description>
cnt<decisionNumber> = 0
<decls>
<@preloop()>
<@loopBody>
<positiveClosureBlockLoop()>
<@end>
<@postloop()>
>>

positiveClosureBlockLoop() ::= <<
while True: #loop<decisionNumber>
   alt<decisionNumber> = <maxAlt>
   <@predecision()>
   <@decisionBody><decision><@end>
   <@postdecision()>
   <alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
   else:
       if cnt<decisionNumber> >= 1:
           break #loop<decisionNumber>

       <ruleBacktrackFailure()>
       eee = EarlyExitException(<decisionNumber>, self.input)
       <@earlyExitException()>
       raise eee

   cnt<decisionNumber> += 1
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
<decls>
<@preloop()>
<@loopBody>
<closureBlockLoop()>
<@end>
<@postloop()>
```

```
>>

closureBlockLoop()
 ::= <<
while True: #loop<decisionNumber>
   alt<decisionNumber> = <maxAlt>
   <@predecision()>
   <@decisionBody><decision><@end>
   <@postdecision()>
   <alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
   else:
       break #loop<decisionNumber>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
if alt<decisionNumber> == <altNum>:
   <@prealt()>
   <alt>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,
 treeLevel,rew) ::= <<
# <fileName>:<description>
pass <! so empty alternatives are a valid block !>
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""
```

```
// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)><label> = <endif>self.match(self.input, <token>,
self.FOLLOW_<token>_in_<ruleName><elementIndex>)
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label, elem) ::= <<
if list_<label> is None:
   list_<label> = []
list_<label>.append(<elem>)<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label>
 = self.input.LA(1)<\n>
<endif>
self.match(<char>)
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchRange(<a>, <b>)
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
```

```
<endif>
if <s>:
   self.input.consume()
   <postmatchCode>
<if(!LEXER)>
   self._state.errorRecovery = False<\n>
<endif>

else:
   <ruleBacktrackFailure()>
   mse = MismatchedSetException(None, self.input)
   <@mismatchedSetException()>
<if(LEXER)>
   self.recover(mse)
   raise mse
<else>
   raise mse
   <! use following code to make it recover inline; remove throw mse;
   self.recoverFromMismatchedSet(
       self.input, mse, self.FOLLOW_set_in_<ruleName><elementIndex>
       )
   !>
<endif>
<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label,
 label)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
<label>Start = self.getCharIndex()
self.match(<string>)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(input=self.input, type=INVALID_TOKEN_TYPE, channel=DEFAULT_CHANNEL,
start=<label>Start, stop=self.getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
self.match(<string>)
<endif>
```

```
>>

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
self.matchAny(self.input)
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<wildcard(...)>
<listLabel(label,label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchAny()
>>

wildcardCharListLabel(label,
 elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values. The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
self._state.following.append(self.FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label> = <endif>self.<if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=",
">)<\n>
self._state.following.pop()
>>

/** ids+=rule */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

/** A lexer rule reference
```

```
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access
 full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
self.<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(
   input=self.input,
   type=INVALID_TOKEN_TYPE,
   channel=DEFAULT_CHANNEL,
   start=<label>Start<elementIndex>,
   stop=self.getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
self.<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
<label>Start<elementIndex>
 = self.getCharIndex()
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
self.match(EOF)
<label> = <labelType>(input=self.input, type=EOF, channel=DEFAULT_CHANNEL,
start=<label>Start<elementIndex>, stop=self.getCharIndex()-1)
<label>.setLine(<label>StartLine<elementIndex>)
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>)
<else>
self.match(EOF)
<endif>
>>

// used for left-recursive rules
```

```
recRuleDefArg()                    ::= "<recRuleArg()>"
recRuleArg()                       ::= "_p"
recRuleAltPredicate(ruleName, opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()           ::= "root_0 = $<ruleName>_primary.tree"
recRuleSetReturnAction(src, name)     ::= "$<name> = $<src>.<name>"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if
 self.input.LA(1) == DOWN:
   self.match(self.input, DOWN, None)
   <children:element()>
   self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if not (<evalPredicate(pred, description)>):
   <ruleBacktrackFailure()>
   raise FailedPredicateException(self.input, "<ruleName>", "<description>")

>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel">
else:
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
```

```
  <@noViableAltException()>
   raise nvae<\n>
<endif>
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel"><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>
 = <eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
else:
   alt<decisionNumber> = <eotPredictsAlt>
<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if (<labelExpr>) <if(predicates)>and (<predicates>)<endif>:
   <targetState>
>>
```

```
// Fixed DFA (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<!
 FIXME: this is one of the few occasion, where I miss a switch statement
 in Python.
 ATM this is implemented as a list of if .. elif ..
 This may be replaced by faster a dictionary lookup, when I find a solution
 for the cases when an edge is not a plain dfaAcceptState.
!>
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
else:
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
   <@noViableAltException()>
   raise nvae<\n>
<endif>

>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
<if(eotPredictsAlt)>
else:
   alt<decisionNumber> = <eotPredictsAlt>
<endif>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
if <labels:{it | LA<decisionNumber>
 == <it>}; separator=" or ">:
   <targetState>
>>

// Cyclic DFA
```

```
/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = self.dfa<decisionNumber>.predict(self.input)
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
# lookup tables for DFA #<dfa.decisionNumber>

DFA<dfa.decisionNumber>_eot = DFA.unpack(
   u"<dfa.javaCompressedEOT; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_eof
 = DFA.unpack(
   u"<dfa.javaCompressedEOF; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_min = DFA.unpack(
   u"<dfa.javaCompressedMin; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_max = DFA.unpack(
   u"<dfa.javaCompressedMax; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_accept = DFA.unpack(
   u"<dfa.javaCompressedAccept; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_special = DFA.unpack(
   u"<dfa.javaCompressedSpecial; wrap="\"\n    u\"">"
   )


DFA<dfa.decisionNumber>_transition = [
   <dfa.javaCompressedTransition:{s|DFA.unpack(u"<s; wrap="\"\nu\"">")}; separator=",\n">
]

# class definition for DFA #<dfa.decisionNumber>

class DFA<dfa.decisionNumber>(DFA):
   pass

   <@errorMethod()>

<if(dfa.specialStateSTs)>
   def specialStateTransition(self_, s, input):
       # convince pylint that my self_ magic is ok ;)
       # pylint: disable-msg=E0213

       # pretend we are a member of the recognizer
       # thus semantic predicates
 can be evaluated
       self = self_.recognizer

       _s = s

       <dfa.specialStateSTs:{state | if s == <i0>: <! compressed special state numbers 0..n-1 !>
   <state>}; separator="\nel">

<if(backtracking)>
       if self._state.backtracking > 0:
           raise BacktrackingFailed

<endif>
       nvae = NoViableAltException(self_.getDescription(), <dfa.decisionNumber>, _s, input)
       self_.error(nvae)
       raise nvae<\n>
<endif>

>>

cyclicDFAInit(dfa) ::= <<
self.dfa<dfa.decisionNumber> = self.DFA<dfa.decisionNumber>(
   self, <dfa.decisionNumber>,
   eot = self.DFA<dfa.decisionNumber>_eot,
   eof = self.DFA<dfa.decisionNumber>_eof,
   min = self.DFA<dfa.decisionNumber>_min,
   max = self.DFA<dfa.decisionNumber>_max,
   accept = self.DFA<dfa.decisionNumber>_accept,
   special = self.DFA<dfa.decisionNumber>_special,
   transition = self.DFA<dfa.decisionNumber>_transition
   )<\n>
>>
```

```
/** A state in a cyclic DFA; it's a special state and part of a big switch
 on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
index<decisionNumber>_<stateNumber> = input.index()
input.rewind()<\n>
<endif>
s = -1
<edges; separator="\nel">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if s >= 0:
   return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if (<labelExpr>)<if(predicates)> and (<predicates>)<endif>:
   s = <targetStateNumber><\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
se:
   s = <targetStateNumber><\n>
>>


// D
 F A  E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "(<operands; separator=\" or \">)"

notPredicate(pred) ::= "not (<evalPredicate(pred, {})>)"

evalPredicate(pred,description) ::= "(<pred>)"
```

```
evalSynPredicate(pred,description) ::= "self.<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "self.input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(<lower> \<= LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(<lower> \<=
self.input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= "<ranges; separator=\" or \">"

// A T T R I B U T E S

globalAttributeScopeClass(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
class
 <scope.name>_scope(object):
   def __init__(self):
       <scope.attributes:{it | self.<it.decl> = None}; separator="\n">

<endif>
<endif>
>>

globalAttributeScopeStack(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
<endif>
>>

ruleAttributeScopeClass(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
class <scope.name>_scope(object):
   def __init__(self):
       <scope.attributes:{it | self.<it.decl> = None}; separator="\n">

<endif>
```

```
<endif>
>>

ruleAttributeScopeStack(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
<endif>
>>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> = None
>>

returnStructName(r) ::= "<r.name>_return"

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave
 a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
class <ruleDescriptor:returnStructName()>(<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope):
   def __init__(self):
       super(<grammar.recognizerName>.<ruleDescriptor:returnStructName()>, self).__init__()

       <if(scope)><scope.attributes:{it | self.<it.decl> = None}; separator="\n"><endif>
       <@ruleReturnInit()>


   <@ruleReturnMembers()>

<endif>
>>

parameterScope(scope) ::= <<
<if(scope)><scope.attributes:{it | <it.decl>}; separator=", "><endif>
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>"
```

```
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
self.<scope>_stack[-<negIndex>].<attr.name>
<else>
<if(index)>
self.<scope>_stack[<index>].<attr.name>
<else>
self.<scope>_stack[-1].<attr.name>
<endif>
<endif>
%>

/* not applying patch because of bug in action parser!

<if(negIndex)>
((len(self.<scope>_stack)
 - <negIndex> - 1) >= 0 and [self.<scope>_stack[-<negIndex>].<attr.name>] or [None])[0]
<else>
<if(index)>
((<index> \< len(self.<scope>_stack)) and [self.<scope>_stack[<index>].<attr.name>] or [None])[0]
<else>
((len(self.<scope>_stack) > 0) and [self.<scope>_stack[-1].<attr.name>] or [None])[0]
<endif>
<endif>

*/

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<!FIXME: this seems not to be used by ActionTranslator...!>
self.<scope>_stack[-<negIndex>].<attr.name> = <expr>
<else>
<if(index)>
<!FIXME: this seems not to be used by ActionTranslator...!>
self.<scope>_stack[<index>].<attr.name> = <expr>
<else>
self.<scope>_stack[-1].<attr.name> = <expr>
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "self.<scope>_stack"
```

```
/**
 reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
((<scope> is not None) and [<scope>.<attr.name>] or [None])[0]
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> = <expr>
<else>
<attr.name> = <expr>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>.text"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.type"
tokenLabelPropertyRef_line(scope,attr)
 ::= "<scope>.line"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.charPositionInLine"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.channel"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.index"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
```

```
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
((<scope> is not None) and [self.input.getTokenStream().toString(
   self.input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
   self.input.getTreeAdaptor().getTokenStopIndex(<scope>.start)
   )] or [None])[0]
<else>
((<scope> is not None) and [self.input.toString(<scope>.start,<scope>.stop)] or [None])[0]
<endif>
%>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> is not None) and [<scope>.st] or [None])[0]"

/** Isolated $RULE ref ok in lexer as
 it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> is not None) and [<scope>.type] or [0])[0]"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> is not None) and [<scope>.line] or [0])[0]"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> is not None) and [<scope>.charPositionInLine] or
[0])[0]"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "((<scope> is not None) and [<scope>.channel] or [0])[0]"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "((<scope> is not None) and [<scope>.index] or [0])[0]"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> is not None) and [<scope>.text] or [None])[0]"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "((<scope> is not None) and [int(<scope>.text)] or [0])[0]"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop" //mmm... or input.LT(-1)??
rulePropertyRef_tree(scope,attr)
 ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= "self.input.toString(retval.start, self.input.LT(-1))"
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "self._state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "self._state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "self._state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(self.getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"
```

```
/**
 How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if <actions.(actionScope).synpredgate>:
   pass
   <action>

<else>
if <actions.(actionScope).synpredgate>:
   pass
   <action>

<endif>
<else>
#action start
<action>
#action end
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"



// M I S C (properties, etc...)

codeFileExtension() ::= ".py"

true_value() ::= "True"
false_value() ::= "False"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/Python.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
/** Add a variable to track last
 element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _last;<\n>
<ASTLabelType> _first_0;<\n>
>>

/** Add a variable to track last element matched */
ruleInitializations() ::= <<
<super.ruleInitializations()>
_last = NULL;<\n>
_first_0 = NULL;<\n>
>>
```

```
/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( ADAPTOR->getParent(ADAPTOR, retval.tree) != NULL && ADAPTOR->isNilNode(ADAPTOR,
ADAPTOR->getParent(ADAPTOR, retval.tree) ) )
{
  retval.tree = (<ASTLabelType>)ADAPTOR->getParent(ADAPTOR, retval.tree);
}
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
   enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)LT(1);
{
<ASTLabelType>
 _save_last_<treeLevel>;
<ASTLabelType> _first_<treeLevel>;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel>;
<endif>
_save_last_<treeLevel> = _last;
_first_<treeLevel>     = NULL;
<if(!rewriteMode)>
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel> == NULL ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel> == NULL ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<if(backtracking)>}<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( LA(1)==ANTLR3_TOKEN_DOWN ) {
   MATCHT(ANTLR3_TOKEN_DOWN, NULL);
   <children:element()>
   MATCHT(ANTLR3_TOKEN_UP, NULL);
}
<else>
MATCHT(ANTLR3_TOKEN_DOWN, NULL);
<children:element()>
MATCHT(ANTLR3_TOKEN_UP, NULL);
<endif>
<if(!rewriteMode)>
ADAPTOR->addChild(ADAPTOR, root_<enclosingTreeLevel>,
 root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
if ( _first_<treeLevel> == NULL ) _first_<treeLevel> = <label>;
<if(backtracking)>}<endif>
<endif>
>>

/** label+=TOKEN auto construct */
```

```
tokenRefAndListLabel(token,label,elementIndex,terminalOptions)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate>
 ) {<endif>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupTree(ADAPTOR, <label>);
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)LT(1);
```

```
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif><\n>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>\}<endif>
<endif>
}
,...)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode)
 ::= <<
_last = (<ASTLabelType>)LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ){<endif>
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif>
root_<treeLevel> = (<ASTLabelType>)ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_<treeLevel>);
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRef(...)>
```

```
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )
{
<endif>
<if(!rewriteMode)>
 ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>.tree);
<else>
 if ( _first_<treeLevel> == NULL ) _first_<treeLevel> = <label>.tree;
<endif>
<if(backtracking)>}<endif>
>>

/**
 x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<super.listLabelAST(elem=label,...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( ( <actions.(actionScope).synpredgate> ) ) <endif>root_<treeLevel> =
(<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>.tree, root_<treeLevel>));
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<super.listLabelAST(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefTrackAndListLabel(...)>
>>
/**
 ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
```

```
_last = (<ASTLabelType>)LT(1);
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>



/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>New(stream_<token>->nextNode(stream_<token>))
<else>
stream_<token>->nextNode(stream_<token>)
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp(...)>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<if(!ruleDescriptor.isSynPred)>
 retval.stop = LT(-1);<\n>
<endif>
 retval.tree = (<ASTLabelType>)ADAPTOR->rulePostProcessing(ADAPTOR, root_0);
<if(backtracking)>
}
<endif>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
    :{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/ASTTreeParser.stg

No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2012 Terence Parr

```
/** The API version of the runtime that recognizers generated by this runtime
*  need.
*/
apiVersion() ::= "1"

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
           bitsets, buildTemplate, buildAST, rewriteMode, profile,
           backtracking, synpreds, memoize,
 numRules,
           fileName, ANTLRVersion, generatedTimestamp, trace,
           scopes, superClass, literals) ::=
<<
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
import sys
from antlr3 import *
```

```
<if(TREE_PARSER)>
from antlr3.tree import *<\n>
<endif>
<@end>

<actions.(actionScope).header>

<! <docComment> !>

# for convenience in actions
HIDDEN = BaseRecognizer.HIDDEN

# token types
<tokens:{it | <it.name>=<it.type>}; separator="\n">

# token names
tokenNamesMap = {
   0: "\<invalid>", 1: "\<EOR>", 2: "\<DOWN>", 3: "\<UP>",
   <tokens:{it | <it.type>: "<it.name>"}; wrap, separator=", ">
}
Token.registerTokenNamesMap(tokenNamesMap)

<recognizer>

<if(actions.(actionScope).main)>
<actions.(actionScope).main>
<else>
def main(argv, stdin=sys.stdin, stdout=sys.stdout, stderr=sys.stderr):
<if(LEXER)>
   from antlr3.main import LexerMain
   main = LexerMain(<recognizer.name>)<\n>
<endif>
<if(PARSER)>
   from antlr3.main import ParserMain
   main = ParserMain("<recognizer.grammar.name>Lexer",
 <recognizer.name>)<\n>
<endif>
<if(TREE_PARSER)>
   from antlr3.main import WalkerMain
   main = WalkerMain(<recognizer.name>)<\n>
<endif>
   main.stdin = stdin
   main.stdout = stdout
   main.stderr = stderr
   main.execute(argv)<\n>
<endif>

<actions.(actionScope).footer>
```

```
if __name__ == '__main__':
   main(sys.argv)

>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode,
     labelType="CommonToken", superClass="Lexer") ::= <<
<if(grammar.directDelegates)>
# path hack to allow absolute import of related grammars.
from os.path import dirname
__path__ = [dirname(__file__)]
del dirname

<grammar.directDelegates:
{g|from .<g.recognizerName> import <g.recognizerName>}; separator="\n">
<endif>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
   <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>

   grammarFileName = "<fileName>"
   api_version = <apiVersion()>

   def __init__(self<grammar.delegators:{g|,
 <g:delegateName()>}>, input=None, state=None):
       if state is None:
           state = RecognizerSharedState()
       super().__init__(input, state)

<if(memoize)>
<if(grammar.grammarIsRoot)>
       self._state.ruleMemo = {}
<endif>
<endif>

       <grammar.directDelegates:
        {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)}; separator="\n">
       <grammar.directDelegates:
        {g|<g.delegates:{h|self.<h:delegateName()> = self.<g:delegateName()>.<h:delegateName()>};
separator="\n">}; separator="\n">
       <grammar.delegators:
        {g|self.<g:delegateName()> = <g:delegateName()>}; separator="\n">
       <last(grammar.delegators):
        {g|self.gParent = <g:delegateName()>}; separator="\n">
       self.delegates = [<grammar.delegates: {g|self.<g:delegateName()>}; separator = ", ">]
```

```
        <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">



 <actions.lexer.init>



   <actions.lexer.members>



<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>



>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
def nextToken(self):
   while True:
       if self.input.LA(1) == EOF:
           return self.makeEOFToken()

       self._state.token = None
       self._state.channel = DEFAULT_CHANNEL
       self._state.tokenStartCharIndex = self.input.index()
       self._state.tokenStartCharPositionInLine = self.input.charPositionInLine
       self._state.tokenStartLine = self.input.line
       self._state._text
 = None
       try:
           m = self.input.mark()
           try:
               # means we won't throw slow exception
               self._state.backtracking = 1
               try:
                   self.mTokens()
               finally:
                   self._state.backtracking = 0

           except BacktrackingFailed:
               # mTokens backtracks with synpred at backtracking==2
               # and we set the synpredgate to allow actions at level 1.
               self.input.rewind(m)
               self.input.consume() # advance one char and try again

           else:
               self.emit()
               return self._state.token

       except RecognitionException as re:
           # shouldn't happen in backtracking mode, but...
           self.reportError(re)
           self.recover(re)


def memoize(self, input, ruleIndex, ruleStartIndex, success):
   if self._state.backtracking > 1:
       # is Lexer always superclass?

 super().memoize(input, ruleIndex, ruleStartIndex, success)


def alreadyParsedRule(self, input, ruleIndex):
   if self._state.backtracking > 1:
       return super().alreadyParsedRule(input, ruleIndex)
   return False


>>

actionGate() ::= "self._state.backtracking == 0"

filteringActionGate() ::= "self._state.backtracking == 1"

/** How to generate a parser */

genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
              bitsets, inputStreamType, superClass, labelType, members,
              rewriteElementType, filterMode, init, ASTLabelType="Object") ::= <<
# token names
tokenNames = [
   "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>",
   <tokenNames; wrap, separator=", ">
]
```

```
<scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScopeClass(scope=it)><endif>}>

<if(grammar.directDelegates)>
# path hack to allow absolute import of related grammars.
from os.path import dirname
__path__ = [dirname(__file__)]
del dirname

<grammar.directDelegates:
{g|from
 .<g.recognizerName> import <g.recognizerName>}; separator="\n">
<endif>

<rules:{it|<ruleAttributeScopeClass(scope=it.ruleDescriptor.ruleScope)>}>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
   grammarFileName = "<fileName>"
   api_version = <apiVersion()>
   tokenNames = tokenNames

   def __init__(self<grammar.delegators:{g|, <g:delegateName()>}>, input, state=None, *args, **kwargs):
       if state is None:
           state = RecognizerSharedState()

       <@args()>
       super().__init__(input, state, *args, **kwargs)

<if(memoize)>
<if(grammar.grammarIsRoot)>
       self._state.ruleMemo = {}
<endif>
<endif>

       <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">

       <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeStack(scope=it)><endif>}>
       <rules:{it | <ruleAttributeScopeStack(scope=it.ruleDescriptor.ruleScope)>}>

       <init>

       <grammar.delegators:
        {g|self.<g:delegateName()> = <g:delegateName()>};
 separator="\n">
       <grammar.directDelegates:
        {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)}; separator="\n">
       <grammar.directDelegates:
        {g|<g.delegates:{h|self.<h:delegateName()> = self.<g:delegateName()>.<h:delegateName()>};
separator="\n">}; separator="\n">
       <last(grammar.delegators):
        {g|self.gParent = self.<g:delegateName()>}; separator="\n">
       self.delegates = [<grammar.delegates: {g|self.<g:delegateName()>}; separator = ", ">]

       <@init><@end>


   <@members><@end>

   <members>

   <rules; separator="\n\n">

   <! generate rule/method definitions for imported rules so they
      appear to be defined in this recognizer. !>
   <grammar.delegatedRules:{ruleDescriptor| <delegateRule(ruleDescriptor)> }; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:{it | FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>
 = frozenset([<it.tokenTypes:{it | <it>};separator=", ">])<\n>}>

>>

delegateRule(ruleDescriptor) ::= <<
def <ruleDescriptor.name>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<\ >   <if(ruleDescriptor.hasReturnValue)>return
<endif>self.<ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<if(ruleDescriptor.parameterScope)
><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", "><endif>)



>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
      ASTLabelType="Object", superClass="Parser", labelType="Token",
      members={<actions.parser.members>},
      init={<actions.parser.init>}
      ) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, "TokenStream", superClass,
            labelType, members, "Token",
            false, init, ASTLabelType)>
>>

/** How to generate a tree parser; same as parser except the input
```

```
*  stream
 is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
          numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
 members={<actions.treeparser.members>},
          init={<actions.treeparser.init>}
          ) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, "TreeNodeStream", superClass,
            labelType, members, "Node",
            filterMode, init, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule,
 they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start "<ruleName>"
def <ruleName>_fragment(self, <ruleDescriptor.parameterScope:parameterScope()>):
   <ruleLabelDefs()>
<if(trace)>
   self.traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
   try:
       <block>

   finally:
       self.traceOut("<ruleName>_fragment", <ruleDescriptor.index>)

<else>
   <block>
<endif>
# $ANTLR end "<ruleName>"



>>

synpred(name) ::= <<
def <name>(self):
   self._state.backtracking += 1
   <@start()>
```

```
    start = self.input.mark()
    try:
        self.<name>_fragment()
    except BacktrackingFailed:
        success = False
    else:
        success = True
    self.input.rewind(start)
    <@stop()>
    self._state.backtracking -= 1
    return success


>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if self._state.backtracking > 0 and self.alreadyParsedRule(self.input,
 <ruleDescriptor.index>):
    # for cached failed rules, alreadyParsedRule will raise an exception
    success = True
    return <ruleReturnValue()>

<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if self._state.backtracking > 0:
    raise BacktrackingFailed

<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<returnScope(scope=ruleDescriptor.returnScope)>

# $ANTLR start "<ruleName>"
# <fileName>:<description>
<ruleDescriptor.actions.decorate>
```

```
def <ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<if(trace)>
   self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>

 <@body><ruleBody()><@end>
   <@postamble()>
   return <ruleReturnValue()>

# $ANTLR end "<ruleName>"
>>

ruleBody() ::= <<
<if(memoize)>
<if(backtracking)>
success = False<\n>
<endif>
<endif>
try:
   try:
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>

<if(memoize)>
<if(backtracking)>
       success = True<\n>
<endif>
<endif>
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   except RecognitionException as re:
       self.reportError(re)
       self.recover(self.input, re)
       <@setErrorReturnValue()>

<endif>
```

```
<else>
  finally:
      pass

<endif>
<endif>
finally:
<if(trace)>
  self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
  pass
>>

catch(decl,action) ::= <<
except <e.decl>:
   <e.action>

>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval = self.<ruleDescriptor.name>_return()
retval.start = self.input.LT(1)<\n>
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name> = <if(a.initValue)><a.initValue><else>None<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex = self.input.index()
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it | self.<it>_stack.append(<it>_scope())}; separator="\n">
<ruleDescriptor.ruleScope:{it | self.<it.name>_stack.append(<it.name>_scope())}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it | self.<it>_stack.pop()}; separator="\n">
<ruleDescriptor.ruleScope:{it | self.<it.name>_stack.pop()}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it | <it.label.text> = None}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,
  ruleDescriptor.wildcardTreeListLabels]
   :{it | list_<it.label.text> = None}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{it | <it.label.text> = None}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it | <it.label.text> = None}; separator="\n"
>
<ruleDescriptor.charLabels:{it | <it.label.text> = None}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{it | list_<it.label.text> = None}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop
 = self.input.LT(-1)<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
```

```
if self._state.backtracking > 0:
   self.memoize(self.input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex, success)

<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start "<ruleName>"
def m<ruleName>(self, <ruleDescriptor.parameterScope:parameterScope()>):
<if(trace)>
   self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
<if(memoize)>
<if(backtracking)>
   success = False<\n>
<endif>
<endif>
   try:
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       _type = <ruleName>
       _channel = DEFAULT_CHANNEL

       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
        <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       self._state.type = _type
       self._state.channel = _channel
       <(ruleDescriptor.actions.after):execAction()>
<endif>
<if(memoize)>
<if(backtracking)>
       success = True<\n>
<endif>
<endif>
```

```
   finally:
<if(trace)>
       self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
       <ruleScopeCleanUp()>
       <memoize()>
       pass

# $ANTLR end "<ruleName>"



>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
def mTokens(self):
   <block><\n>



>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@body><blockBody()><@end>
>>

blockBody() ::= <<
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<@prebranch()>
<alts:{a
 | <altSwitchCase(i, a)>}; separator="\nel">
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
```

```
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<
# <fileName>:<description>
cnt<decisionNumber> = 0
<decls>
<@preloop()>
<@loopBody>
<positiveClosureBlockLoop()>
<@end>
<@postloop()>
>>

positiveClosureBlockLoop() ::= <<
while True: #loop<decisionNumber>
  alt<decisionNumber> = <maxAlt>
  <@predecision()>
  <@decisionBody><decision><@end>
  <@postdecision()>
  <alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
  else:
    if cnt<decisionNumber> >= 1:
      break #loop<decisionNumber>
```

```
        <ruleBacktrackFailure()>
        eee = EarlyExitException(<decisionNumber>, self.input)
        <@earlyExitException()>
        raise eee

   cnt<decisionNumber> += 1
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
<decls>
<@preloop()>
<@loopBody>
<closureBlockLoop()>
<@end>
<@postloop()>
>>

closureBlockLoop()
 ::= <<
while True: #loop<decisionNumber>
   alt<decisionNumber> = <maxAlt>
   <@predecision()>
   <@decisionBody><decision><@end>
   <@postdecision()>
   <alts:{a | <altSwitchCase(i, a)>}; separator="\nel">
   else:
       break #loop<decisionNumber>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
```

```
if alt<decisionNumber> == <altNum>:
   <@prealt()>
   <alt>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt, treeLevel,rew) ::= <<
#
 <fileName>:<description>
pass <! so empty alternatives are a valid block !>
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)><label> = <endif>self.match(self.input, <token>,
self.FOLLOW_<token>_in_<ruleName><elementIndex>)
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label, elem) ::= <<
if list_<label> is None:
   list_<label> = []
list_<label>.append(<elem>)<\n>
>>

/** match a character */
```

```
charRef(char,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.match(<char>)
>>

/**
 match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchRange(<a>, <b>)
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
if <s>:
   self.input.consume()
   <postmatchCode>
<if(!LEXER)>
   self._state.errorRecovery = False<\n>
<endif>

else:
   <ruleBacktrackFailure()>
   mse = MismatchedSetException(None, self.input)
   <@mismatchedSetException()>
<if(LEXER)>
   self.recover(mse)
   raise mse
<else>
   raise mse
   <! use following code to make it recover inline; remove throw mse;
   self.recoverFromMismatchedSet(
       self.input, mse, self.FOLLOW_set_in_<ruleName><elementIndex>
       )
   !>
<endif>
<\n>
>>

matchRuleBlockSet ::= matchSet
```

```
matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

/**
 Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
<label>Start = self.getCharIndex()
self.match(<string>)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(input=self.input, type=INVALID_TOKEN_TYPE, channel=DEFAULT_CHANNEL,
start=<label>Start, stop=self.getCharIndex()-1)
<label>.line = <label>StartLine<elementIndex>
<label>.charPositionInLine = <label>StartCharPos<elementIndex>
<else>
self.match(<string>)
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
self.matchAny()
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<wildcard(...)>
<listLabel(label,label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchAny()
>>

wildcardCharListLabel(label, elementIndex) ::=
 <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>
```

```
/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values. The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
self._state.following.append(self.FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label> = <endif>self.<if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", 
">)<\n>
self._state.following.pop()
>>

/** ids+=rule */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

/** A lexer rule reference
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope)
 ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
self.<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<label>StartLine<elementIndex> = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
<label> = <labelType>(
   input=self.input,
   type=INVALID_TOKEN_TYPE,
   channel=DEFAULT_CHANNEL,
   start=<label>Start<elementIndex>,
   stop=self.getCharIndex()-1)
<label>.line = <label>StartLine<elementIndex>
<label>.charPositionInLine = <label>StartCharPos<elementIndex>
<else>
self.<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
```

```
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
<label>StartLine<elementIndex>
 = self.getLine()
<label>StartCharPos<elementIndex> = self.getCharPositionInLine()
self.match(EOF)
<label> = <labelType>(input=self.input, type=EOF, channel=DEFAULT_CHANNEL,
start=<label>Start<elementIndex>, stop=self.getCharIndex()-1)
<label>.line = <label>StartLine<elementIndex>
<label>.charPositionInLine = <label>StartCharPos<elementIndex>
<else>
self.match(EOF)
<endif>
>>

// used for left-recursive rules
recRuleDefArg()                       ::= "<recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName, opPrec) ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0 = $<ruleName>_primary.tree"
recRuleSetReturnAction(src, name)     ::= "$<name> = $<src>.<name>"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if self.input.LA(1) == DOWN:
   self.match(self.input,
 DOWN, None)
   <children:element()>
   self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
```

```
validateSemanticPredicate(pred,description) ::= <<
if not (<evalPredicate(pred, description)>):
   <ruleBacktrackFailure()>
   raise FailedPredicateException(self.input, "<ruleName>", "<description>")

>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel">
else:
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
   <@noViableAltException()>
   raise nvae<\n>
<endif>
>>

/**
 Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel"><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>
 = <eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
```

```
else:
   alt<decisionNumber> = <eotPredictsAlt>
<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if (<labelExpr>) <if(predicates)>and (<predicates>)<endif>:
   <targetState>
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<!
 FIXME: this is one of the few occasion, where I miss a switch statement
 in Python. ATM this is implemented as a list of if .. elif ..
 This
 may be replaced by faster a dictionary lookup, when I find a solution
 for the cases when an edge is not a plain dfaAcceptState.
!>
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
else:
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
   <@noViableAltException()>
   raise nvae<\n>
<endif>

>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
```

```
<edges; separator="\nel">
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
<if(eotPredictsAlt)>
else:
   alt<decisionNumber> = <eotPredictsAlt>
<endif>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
if LA<decisionNumber> in {<labels; separator=", ">}:
   <targetState>
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = self.dfa<decisionNumber>.predict(self.input)
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
# lookup tables for DFA #<dfa.decisionNumber>

DFA<dfa.decisionNumber>_eot = DFA.unpack(
   "<dfa.javaCompressedEOT; wrap="\"\n    \"">"
   )

DFA<dfa.decisionNumber>_eof = DFA.unpack(
   "<dfa.javaCompressedEOF; wrap="\"\n    \"">"
    )

DFA<dfa.decisionNumber>_min = DFA.unpack(
   "<dfa.javaCompressedMin; wrap="\"\n    \"">"
   )

DFA<dfa.decisionNumber>_max = DFA.unpack(
   "<dfa.javaCompressedMax; wrap="\"\n    \"">"
   )

DFA<dfa.decisionNumber>_accept = DFA.unpack(
   "<dfa.javaCompressedAccept; wrap="\"\n    \"">"
   )

DFA<dfa.decisionNumber>_special = DFA.unpack(
   "<dfa.javaCompressedSpecial; wrap="\"\n    \"">"
   )


DFA<dfa.decisionNumber>_transition = [
   <dfa.javaCompressedTransition:{s|DFA.unpack("<s; wrap="\"\n\"">")}; separator=",\n">
]

# class definition for DFA #<dfa.decisionNumber>

class DFA<dfa.decisionNumber>(DFA):
   pass

   <@errorMethod()>

<if(dfa.specialStateSTs)>
   def specialStateTransition(self_, s, input):
       # convince pylint that my self_ magic is ok ;)
       # pylint: disable-msg=E0213

       # pretend we are a member of the recognizer
       # thus semantic predicates can be evaluated
       self = self_.recognizer

       _s = s

       <dfa.specialStateSTs:{state
  | if s == <i0>: <! compressed special state numbers 0..n-1 !>
   <state>}; separator="\nel">

<if(backtracking)>
       if self._state.backtracking > 0:
           raise BacktrackingFailed

<endif>
       nvae = NoViableAltException(self_.getDescription(), <dfa.decisionNumber>, _s, input)
```

```
        self_.error(nvae)
        raise nvae<\n>
<endif>

>>

cyclicDFAInit(dfa) ::= <<
self.dfa<dfa.decisionNumber> = self.DFA<dfa.decisionNumber>(
   self, <dfa.decisionNumber>,
   eot = self.DFA<dfa.decisionNumber>_eot,
   eof = self.DFA<dfa.decisionNumber>_eof,
   min = self.DFA<dfa.decisionNumber>_min,
   max = self.DFA<dfa.decisionNumber>_max,
   accept = self.DFA<dfa.decisionNumber>_accept,
   special = self.DFA<dfa.decisionNumber>_special,
   transition = self.DFA<dfa.decisionNumber>_transition
   )<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState)
 ::= <<
LA<decisionNumber>_<stateNumber> = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
index<decisionNumber>_<stateNumber> = input.index()
input.rewind()<\n>
<endif>
s = -1
<edges; separator="\nel">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if s >= 0:
   return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if (<labelExpr>)<if(predicates)> and (<predicates>)<endif>:
   s = <targetStateNumber><\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
```

```
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
se:
   s = <targetStateNumber><\n>
>>



// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands)
 ::= "(<operands; separator=\" or \">)"

notPredicate(pred) ::= "not (<evalPredicate(pred, {})>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "self.<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "self.input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(<lower> \<= LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(<lower> \<=
self.input.LA(<k>) \<= <upper>)"

lookaheadSetTest(values,k,valuesAsInt) ::= <%
LA<decisionNumber>_<stateNumber> in {<values; separator=", ">}
%>

isolatedLookaheadSetTest(values,k,valuesAsInt) ::= <%
self.input.LA(<k>) in {<values; separator=", ">}
%>

lookaheadVarName(k)
 ::= "LA<decisionNumber>_<stateNumber>"
isolatedLookaheadVarName(k) ::= "self.input.LA(<k>)"

setTest(ranges) ::= "<ranges; separator=\" or \">"
```

```
// A T T R I B U T E S

globalAttributeScopeClass(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
class <scope.name>_scope(object):
   def __init__(self):
       <scope.attributes:{it | self.<it.decl> = None}; separator="\n">

<endif>
<endif>
>>

globalAttributeScopeStack(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
<endif>
>>

ruleAttributeScopeClass(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
class <scope.name>_scope(object):
   def __init__(self):
       <scope.attributes:{it | self.<it.decl> = None}; separator="\n">

<endif>
<endif>
>>

ruleAttributeScopeStack(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
<endif>
>>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Define a rule label including
 default value */
ruleLabelDef(label) ::= <<
<label.label.text> = None
```

```
>>

returnStructName(r) ::= "<r.name>_return"

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
class <ruleDescriptor:returnStructName()>(<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope):
   def __init__(self):
       super().__init__()

       <if(scope)><scope.attributes:{it | self.<it.decl> = None}; separator="\n"><endif>
       <@ruleReturnInit()>



   <@ruleReturnMembers()>

<endif>
>>

parameterScope(scope) ::= <<
<if(scope)><scope.attributes:{it | <it.decl>}; separator=", "><endif>
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>"

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
self.<scope>_stack[-<negIndex>].<attr.name>
<else>
<if(index)>
self.<scope>_stack[<index>].<attr.name>
<else>
self.<scope>_stack[-1].<attr.name>
<endif>
<endif>
%>

/*
 not applying patch because of bug in action parser!

<if(negIndex)>
((len(self.<scope>_stack) - <negIndex> - 1) >= 0 and [self.<scope>_stack[-<negIndex>].<attr.name>] or [None])[0]
<else>
```

```
<if(index)>
((<index> \< len(self.<scope>_stack)) and [self.<scope>_stack[<index>].<attr.name>] or [None])[0]
<else>
((len(self.<scope>_stack) > 0) and [self.<scope>_stack[-1].<attr.name>] or [None])[0]
<endif>
<endif>

*/

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<!FIXME: this seems not to be used by ActionTranslator...!>
self.<scope>_stack[-<negIndex>].<attr.name> = <expr>
<else>
<if(index)>
<!FIXME: this seems not to be used by ActionTranslator...!>
self.<scope>_stack[<index>].<attr.name> = <expr>
<else>
self.<scope>_stack[-1].<attr.name> = <expr>
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
 *  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "self.<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
((<scope> is not None) and [<scope>.<attr.name>] or [None])[0]
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
```

```
retval.<attr.name> = <expr>
<else>
<attr.name> = <expr>
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach; and they are evaluated early;
// they cannot
 see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>.text"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.type"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.line"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.charPositionInLine"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.channel"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.index"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
((<scope> is not None) and [self.input.getTokenStream().toString(
   self.input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
   self.input.getTreeAdaptor().getTokenStopIndex(<scope>.start)
   )] or [None])[0]
<else>
((<scope> is not None) and [self.input.toString(<scope>.start,<scope>.stop)]
 or [None])[0]
<endif>
%>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> is not None) and [<scope>.st] or [None])[0]"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> is not None) and [<scope>.type] or [0])[0]"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> is not None) and [<scope>.line] or [0])[0]"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> is not None) and [<scope>.charPositionInLine] or
[0])[0]"
```

```
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "((<scope> is not None) and [<scope>.channel] or [0])[0]"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "((<scope> is not None) and [<scope>.index] or [0])[0]"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> is not None) and [<scope>.text] or [None])[0]"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "((<scope> is not None) and [int(<scope>.text)] or [0])[0]"

// Somebody may ref $template
 or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop" //mmm... or input.LT(-1)??
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= "self.input.toString(retval.start, self.input.LT(-1))"
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "self._state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "self._state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "self._state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(self.getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st and $tree is allowed
 in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"


/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if <actions.(actionScope).synpredgate>:
   pass
   <action>

<else>
if <actions.(actionScope).synpredgate>:
   pass
   <action>

<endif>
<else>
#action start
<action>
#action end
```

```
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"



// M I S C (properties, etc...)

codeFileExtension() ::= ".py"

true_value() ::= "True"
false_value() ::= "False"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/Python3.stg
No license file was found, but licenses were detected in source scan.

group Dbg;

/*
[The "BSD license"]

```
*/

@parserBody.mixins() ::= <<
include ANTLR3::<if(profile)>Profile<else>Debug<endif>::ParserEvents
>>


@parserBody.additionalMembers() ::= <<
<if(grammar.grammarIsRoot)>
RULE_NAMES = [
 :invalid_rule, <grammar.allImportedRules:{rST|:<rST.name>}; wrap="\n  ", separator=", ">
].freeze
<endif>
>>


@rule.body() ::= <<
in_rule(grammar_file_name, "<ruleName>") do
 @debug_listener.location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>)

 <@super.body()>

 @debug_listener.location(<ruleDescriptor.EORNode.line>,
 <ruleDescriptor.EORNode.column>)
end
>>


// Common debug event triggers used by region overrides below
enterSubRule() ::= <<
in_subrule(<decisionNumber>) do<\n>
>>


exitSubRule() ::= <<
end<\n>
>>


enterDecision() ::= <<
in_decision(<decisionNumber>) do<\n>
>>


exitDecision() ::= <<
end<\n>
>>
```

```
enterAlt(n) ::= <<
in_alternative(<n>)<\n>
>>


// Region overrides that tell various constructs to add debugging triggers
@block.body() ::= <<
in_subrule(<decisionNumber>) do
 <@super.body()>
end
>>


// @blockBody.predecision() ::= "<enterSubRule()>"
// @blockBody.postdecision() ::= "<exitDecision()>"
// @blockBody.postbranch() ::= "<exitSubRule()>"
@blockBody.decision() ::= <<
in_decision(<decisionNumber>) do
 <@super.decision()>
end
>>


@ruleBlock.decision() ::= <<
in_decision(<decisionNumber>) do
 <@super.decision()>
end<\n>
>>


@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n={1})>"


@blockSingleAlt.prealt() ::= "<enterAlt(n={1})>"


@positiveClosureBlock.loopBody()
 ::= <<
in_subrule(<decisionNumber>) do
 <@super.loopBody()>
end
>>


@positiveClosureBlockLoop.decisionBody() ::= <<
in_decision(<decisionNumber>) do
 <@super.decisionBody()>
```

```
end
>>


@positiveClosureBlockLoop.earlyExitException() ::= <<
@debug_listener.recognition_exception(eee)
>>


@closureBlock.loopBody() ::= <<
in_subrule(<decisionNumber>) do
 <@super.loopBody()>
end
>>


@closureBlockLoop.decisionBody() ::= <<
in_decision(<decisionNumber>) do
 <@super.decisionBody()>
end
>>


@altSwitchCase.prealt() ::= "<enterAlt(altNum)>"  // altNum is arg of altSwitchCase


element(e) ::= <<
@debug_listener.location(<e.line>, <e.pos>) // e is arg of element
<super.element(e)>
>>


@matchSet.mismatchedSetException() ::= "@debug_listener.recognition_exception(mse)"


@dfaState.noViableAltException() ::= <<
nvae = NoViableAlternative( "<description>", <decisionNumber>, <stateNumber> )
@debug_listener.recognition_exception( nvae )
raise( nvae )<\n>
>>

@dfaStateSwitch.noViableAltException()
 ::= <<
nvae = NoViableAlternative( "<description>", <decisionNumber>, <stateNumber> )
@debug_listener.recognition_exception( nvae )
raise( nvae )<\n>
>>
```

```
dfaDecision(decisionNumber,description) ::= <<
begin
 @state.cyclic_decision = true
 <super.dfaDecision(...)>
rescue ANTLR3::Error::NoViableAlternative => nvae
 @debug_listener.recognition_exception(nvae)
 raise
end
>>



@cyclicDFA.errorMethod() ::= <<
def error(nvae)
 @recognizer.debug_listener.recognition_exception(nvae)
 super
end
>>
/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
predicate?("<description>") { <pred> }
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Ruby/Dbg.stg
No license file was found, but licenses were detected in source scan.

```
* IMPLIED
 WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr3.ST;
using Antlr3.ST.Language;
>>

@genericParser.members() ::= <<
<@super.members()>
private StringTemplateGroup _templateGroup = new StringTemplateGroup("<name>Templates",
 typeof(AngleBracketTemplateLexer) );

public StringTemplateGroup TemplateGroup
{
 get { return _templateGroup; }
 set { _templateGroup = value; }
}
>>

ruleReturnBaseType() ::= <%
Template<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<StringTemplate, <labelType>>
%>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem={<label>.Template},elemType="StringTemplate",...)>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if (<actions.(actionScope).synpredgate>)
```

```
{
 <alts:rewriteTemplateAlt(); separator="else ">
 <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
((TokenRewriteStream)input.TokenStream).Replace(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start),
 retval.Template);
<else>
((TokenRewriteStream)input).Replace(
 retval.Start.TokenIndex,
 input.LT(-1).TokenIndex,
 retval.Template);
<endif>
>>

rewriteTemplateAlt(it)
 ::= <<
// <it.description>
<if(it.pred)>
if (<it.pred>)
{
 retval.Template = <it.alt>;
}<\n>
<else>
{
 retval.Template = <it.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
null;
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
```

```
*/
rewriteExternalTemplate(name,args) ::= <%
TemplateGroup.GetInstanceOf("<name>"<optionalArguments(args)>)
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
TemplateGroup.GetInstanceOf(<expr><optionalArguments(args)>)
%>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args,
 template) ::= <%
new StringTemplate(TemplateGroup, "<template>"<optionalArguments(args)>)
%>

optionalArguments(args) ::= <<
<if(args)>,
 new Dictionary\<string, object>() { <args:optionalArgument(); separator=", "> }
<endif>
>>

optionalArgument(it) ::= <<
{"<it.name>", <it.value>}
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>).SetAttribute("<attrName>",<expr>);
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(TemplateGroup,<stringExpr>)
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/ST.stg
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/
parserMembers() ::= <<
protected DebugTreeAdaptor adaptor =
   new DebugTreeAdaptor(null,new CommonTreeAdaptor());
public void setTreeAdaptor(TreeAdaptor adaptor) {
   this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
}
public TreeAdaptor getTreeAdaptor()
 {
   return adaptor;
}<\n>
>>

parserCtorBody() ::= <<
>>
```

```
createListenerAndHandshake() ::= <<
<super.createListenerAndHandshake()>
>>

ctorForPredefinedListener() ::= <<
>>

@rewriteElement.pregen() ::= ""
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/ASTDbg.stg

No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during normal parsing.
*
```

```
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of
 combinations.
*/
group ASTParser;

@rule.setErrorReturnValue() ::= <<
RetVal.Tree := Adaptor.ErrorNode(Input, RetVal.Start as IToken,
 Input.LT(-1), RE) as I<ASTLabelType>;
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
<label>_tree := <createNodeFromToken(...)>;
Adaptor.AddChild(Root[0], <label>_tree);
<if(backtracking)>
end;
<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<label>_tree := <createNodeFromToken(...)>;
Root[0] := Adaptor.BecomeRoot(<label>_tree, Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
```

```
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions)
 ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,terminalOptions,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label
 plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
<super.matchSet(..., postmatchCode={<if(backtracking)>if (State.Backtracking = 0) then
<endif>Adaptor.AddChild(Root[0], <createNodeFromToken(...)>);})>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)
```

```
matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<if(label)>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
<super.matchSet(..., postmatchCode={<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[0] :=
Adaptor.BecomeRoot(<createNodeFromToken(...)>, Root[0]) as I<ASTLabelType>;})>
>>

// RULE REF
 AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Adaptor.AddChild(Root[0], <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[0] := Adaptor.BecomeRoot(<label>.Tree, Root[0])
as I<ASTLabelType>;
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

//
```

```
 WILDCARD AST

wildcard(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
 <label>_tree := Adaptor.CreateNode(<label>) as I<ASTLabelType>;
 Adaptor.AddChild(Root[0], <label>_tree);
<if(backtracking)>
end;
<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
 <label>_tree := Adaptor.CreateNode(<label>) as I<ASTLabelType>;
 Root[0] := Adaptor.BecomeRoot(<label>_tree, Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
>>

createNodeFromToken(label,terminalOptions) ::= <<
<if(terminalOptions.node)>
T<terminalOptions.node>.Create(<label>) <! new MethodNode(IDLabel) !>
<else>
Adaptor.CreateNode(<label>) as I<ASTLabelType>
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>
if (State.Backtracking = 0)
 then
begin<\n>
<endif>
RetVal.Tree := Adaptor.RulePostProcessing(Root[0]) as I<ASTLabelType>;
<if(!TREE_PARSER)>
Adaptor.SetTokenBoundaries(RetVal.Tree, RetVal.Start as IToken, RetVal.Stop as IToken);
```

```
<endif>
<if(backtracking)>
<\n>end;
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Delphi/ASTParser.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/ObjCTarget.java
No license file was found, but licenses were detected in source scan.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/AST.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2010 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR

```
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

javaTypeInitMap ::= [
 "int":"0",
 "long":"0",
 "float":"0.0f",
 "double":"0.0",
 "boolean":"false",
 "byte":"0",
 "short":"0",
 "char":"0",
 default:"null" // anything other than an atomic type
]

// System.Boolean.ToString() returns "True" and "False", but the proper C# literals are "true" and "false"
// The Java version of Boolean returns "true" and "false", so they map to themselves here.
booleanLiteral ::= [
 "True":"true",
 "False":"false",
 "true":"true",
 "false":"false",
 default:"false"
]

/**
 The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
         docComment, recognizer,
         name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>
```

```
<@imports>
import org.antlr.runtime.*;
<if(TREE_PARSER)>
import org.antlr.runtime.tree.*;
<endif>
import java.util.Stack;
import java.util.List;
import java.util.ArrayList;
<if(backtracking)>
import java.util.Map;
import java.util.HashMap;
<endif>
<@end>

<docComment>
@SuppressWarnings("all")
<recognizer>

>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
    superClass="Lexer")
 ::= <<
public class <grammar.recognizerName> extends <@superClassName><superClass><@end> {
 <tokens:{it | public static final int <it.name>=<it.type>;}; separator="\n">
 <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>
 <actions.lexer.members>

 // delegates
 <grammar.delegates:
  {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
 // delegators
 <grammar.delegators:
  {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
 <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>
 public <superClass>[] getDelegates() {
  return new <superClass>[] {<grammar.delegates: {g|<g:delegateName()>}; separator = ", ">};
 }

 public <grammar.recognizerName>() {} <! needed by subclasses !>
 public <grammar.recognizerName>(CharStream input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>) {
  this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>);
 }
 public <grammar.recognizerName>(CharStream
 input, RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>) {
  super(input,state);
<if(memoize)>
```

```
<if(grammar.grammarIsRoot)>
  state.ruleMemo = new HashMap[<numRules>+1];<\n><! index from 1..n !>
<endif>
<endif>
  <grammar.directDelegates:
   {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
  <grammar.delegators:
   {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
 }
 @Override public String getGrammarFileName() { return "<fileName>"; }

<if(filterMode)>
 <filteringNextToken()>
<endif>
 <rules; separator="\n\n">

 <synpreds:{p | <lexerSynpred(p)>}>

 <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
 <cyclicDFAs:cyclicDFA(); separator="\n\n"><! dump tables for all DFA !>

}
>>

/** A override of
 Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
@Override
public Token nextToken() {
 while (true) {
  if ( input.LA(1)==CharStream.EOF ) {
   Token eof = new CommonToken(input,Token.EOF,
          Token.DEFAULT_CHANNEL,
          input.index(),input.index());
   eof.setLine(getLine());
   eof.setCharPositionInLine(getCharPositionInLine());
   return eof;
  }
  state.token = null;
 state.channel = Token.DEFAULT_CHANNEL;
```

```
  state.tokenStartCharIndex = input.index();
  state.tokenStartCharPositionInLine = input.getCharPositionInLine();
  state.tokenStartLine = input.getLine();
 state.text = null;
  try {
   int m = input.mark();
   state.backtracking=1; <! means we won't throw slow exception !>
   state.failed=false;
   mTokens();
   state.backtracking=0;
   <!
 mTokens backtracks with synpred at backtracking==2
    and we set the synpredgate to allow actions at level 1. !>
   if ( state.failed ) {
    input.rewind(m);
    input.consume(); <! advance one char and try again !>
   }
   else {
    emit();
    return state.token;
   }
  }
  catch (RecognitionException re) {
   // shouldn't happen in backtracking mode, but...
   reportError(re);
   recover(re);
  }
 }
}

@Override
public void memoize(IntStream input,
  int ruleIndex,
  int ruleStartIndex)
{
if ( state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

@Override
public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
if ( state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>

actionGate() ::= "state.backtracking==0"

filteringActionGate() ::= "state.backtracking==1"
```

```
/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules,
 numRules,
            bitsets, inputStreamType, superClass,
            labelType, members, rewriteElementType,
            filterMode, ASTLabelType="Object") ::= <<
public class <grammar.recognizerName> extends <@superClassName><superClass><@end> {
<if(grammar.grammarIsRoot)>
 public static final String[] tokenNames = new String[] {
  "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ", wrap="\n\t\t">
 };
<endif>
 <tokens:{it |public static final int <it.name>=<it.type>;}; separator="\n">

 // delegates
 <grammar.delegates: {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
 public <superClass>[] getDelegates() {
  return new <superClass>[] {<grammar.delegates: {g|<g:delegateName()>}; separator = ", ">};
 }

 // delegators
 <grammar.delegators:
  {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
 <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

 <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>

<@members>
 <!
 WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
 public <grammar.recognizerName>(<inputStreamType> input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>) {
  this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>);
 }
 public <grammar.recognizerName>(<inputStreamType> input, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>) {
  super(input, state);
  <parserCtorBody()>
  <grammar.directDelegates:
   {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
  <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
  <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
 }
<@end>
```

```
 @Override public String[] getTokenNames() { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; }
 @Override public String getGrammarFileName()
 { return "<fileName>"; }

 <members>

 <rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
 // Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
 public <returnType(ruleDescriptor)> <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>)
throws <ruleDescriptor.throwsSpec; separator=", "> { <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<if(ruleDescriptor.parameterScope)><ru
leDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", "><endif>); \}}; separator="\n">

 <synpreds:{p | <synpred(p)>}>

 <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
 <cyclicDFAs:cyclicDFA(); separator="\n\n"><! dump tables for all DFA !>

 <bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
      words64=it.bits)>};
 separator="\n">
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new HashMap[<length(grammar.allImportedRules)>+1];<\n><! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
 {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
     ASTLabelType="Object", superClass="Parser", labelType="Token",
     members={<actions.parser.members>}) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
           bitsets, "TokenStream", superClass,
           labelType, members, "Token",
           false, ASTLabelType)>
>>
```

```
/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
          numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
           members={<actions.treeparser.members>}
          ) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
             bitsets, "TreeNodeStream", superClass,
             labelType, members, "Node",
             filterMode, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public final void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>) throws
<ruleDescriptor.throwsSpec:{x|<x>};
 separator=", "> {
 <ruleLabelDefs()>
<if(trace)>
 traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
 try {
  <block>
 }
 finally {
  traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
 }
<else>
 <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public final boolean <name>() {
 state.backtracking++;
 <@start()>
 int start = input.mark();
 try {
  <name>_fragment(); // can never throw exception
 } catch (RecognitionException re) {
  System.err.println("impossible: "+re);
 }
 boolean success = !state.failed;
 input.rewind(start);
 <@stop()>
 state.backtracking--;
 state.failed=false;
 return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) {
<returnStatement(({<ruleReturnValue()>}))> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if
 (state.failed) <returnStatement(({<ruleReturnValue()>}))><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true;
<returnStatement(({<ruleReturnValue()>}))>}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
<if(isPredefinedRewriteRule.(ruleName) && filterMode && buildAST)>
```

```
@Override
<endif>
public final <returnType(ruleDescriptor)> <ruleName>(<ruleDescriptor.parameterScope:parameterScope()>) throws
<ruleDescriptor.throwsSpec:{x|<x>}; separator=", "> {
 <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
 <ruleScopeSetUp()>
 <ruleDeclarations()>
 <ruleLabelDefs()>
 <ruleDescriptor.actions.init>
 <@preamble()>
 try
 {
  <ruleMemoization(name=ruleName)>
  <block>
  <ruleCleanUp()>
  <(ruleDescriptor.actions.after):execAction()>
 }
<if(exceptions)>
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
 <actions.(actionScope).rulecatch>
<else>
 catch (RecognitionException re) {
  reportError(re);
  recover(input,re);
  <@setErrorReturnValue()>
 }
<endif>
<endif>
<endif>
 finally {
  // do for sure before leaving
  <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
 }
 <@postamble()>
 <returnStatement(({<ruleReturnValue()>}), false)>
}
// $ANTLR end "<ruleName>"
>>

returnStatement(returnValue, force=true) ::= <%
<if(!isEmptyString.(returnValue))>
 return <returnValue>;
```

```
<elseif(force)>
 return;
<endif>
%>

catch(decl,action) ::= <<
catch (<e.decl>) {
 <e.action>
}
>>

ruleDeclarations()
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType(ruleDescriptor)> retval = new <returnType(ruleDescriptor)>();
retval.start = input.LT(1);
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.index();
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.pop();}; separator="\n">
>>



ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
 :{it |<labelType> <it.label.text>=null;};
 separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
 :{it |List\<Object> list_<it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|RuleReturnScope <ll.label.text> = null;}; separator="\n">
```

```
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
 :{it |<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
 :{it |List\<Object> list_<it.label.text>=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop
 = input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
```

```
// $ANTLR start "<ruleName>"
public final void m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>) throws RecognitionException
{
 <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
 <ruleScopeSetUp()>
 <ruleDeclarations()>
 try {
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
<else>
  int _type = <ruleName>;
  int _channel = DEFAULT_TOKEN_CHANNEL;
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
  <ruleCleanUp()>
  state.type = _type;
  state.channel
 = _channel;
  <(ruleDescriptor.actions.after):execAction()>
<endif>
 }
 finally {
  // do for sure before leaving
  <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
  <ruleScopeCleanUp()>
  <memoize()>
 }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
@Override
public void mTokens() throws RecognitionException {
 <block>
}
>>

// S U B R U L E S
```

```
/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
 <alts:{a | <altSwitchCase(i,a)>}>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
 <alts:{a | <altSwitchCase(i,a)>}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
loop<decisionNumber>:
while (true) {
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch
 (alt<decisionNumber>) {
 <alts:{a | <altSwitchCase(i,a)>}>
 default :
  if ( cnt<decisionNumber> >= 1 ) break loop<decisionNumber>;
  <ruleBacktrackFailure()>
  EarlyExitException eee = new EarlyExitException(<decisionNumber>, input);
  <@earlyExitException()>
  throw eee;
 }
 cnt<decisionNumber>++;
}
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
loop<decisionNumber>:
while (true) {
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch (alt<decisionNumber>) {
 <alts:{a | <altSwitchCase(i,a)>}>
 default :
  break loop<decisionNumber>;
 }
}
<@postloop()>
>>
```

```
closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
 *  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
case <altNum> :
 <@prealt()>
 <alt>
 break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::=
 <<
<if(label)><label>=(<labelType>)<endif>match(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex
>); <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList\<Object>();
list_<label>.add(<elem>);
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);
<endif>
match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);
<else>
<label>=<castToLabelType("input.LT(1)")>;
<endif>
<endif>
if ( <s>
 ) {
 input.consume();
 <postmatchCode>
<if(!LEXER)>
 state.errorRecovery=false;
<endif>
```

```
 <if(backtracking)>state.failed=false;<endif>
}
else {
 <ruleBacktrackFailure()>
 MismatchedSetException mse = new MismatchedSetException(null,input);
 <@mismatchedSetException()>
<if(LEXER)>
 recover(mse);
 throw mse;
<else>
 throw mse;
 <! use following code to make it recover inline; remove throw mse;
 recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
 !>
<endif>
}
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
int <label>Start = getCharIndex();
match(<string>); <checkRuleBacktrackFailure()>
int <label>StartLine<elementIndex> = getLine();
int <label>StartCharPos<elementIndex> = getCharPositionInLine();
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
 <label>Start, getCharIndex()-1);
<label>.setLine(<label>StartLine<elementIndex>);
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>);
<else>
match(<string>); <checkRuleBacktrackFailure()>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)>
<label>=<castToLabelType("input.LT(1)")>;
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>
```

```
wildcardAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<wildcard(...)>
<listLabel(label, label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule
 name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);
state._fsp--;
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
int <label>StartLine<elementIndex> = getLine();
int <label>StartCharPos<elementIndex> = getCharPositionInLine();
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
 <checkRuleBacktrackFailure()>
<label> = new <labelType>(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<label>.setLine(<label>StartLine<elementIndex>);
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>);
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
int <label>StartLine<elementIndex> = getLine();
int <label>StartCharPos<elementIndex> = getCharPositionInLine();
match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new <labelType>(input, EOF, Token.DEFAULT_CHANNEL, <label>Start<elementIndex>,
getCharIndex()-1);
<label>.setLine(<label>StartLine<elementIndex>);
<label>.setCharPositionInLine(<label>StartCharPos<elementIndex>);
<else>
match(EOF);
 <checkRuleBacktrackFailure()>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()                       ::= "int <recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
```

```
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
 match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
 <children:element()>
 match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input,
 Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(pred,description)>) ) {
 <ruleBacktrackFailure()>
 throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<edges; separator="\nelse ">
<if((!isTrue.(last(edges).labelExpr)) && (!last(edges).predicates))>
else {
<if(eotPredictsAlt)>
 alt<decisionNumber>=<eotPredictsAlt>;
<else>
 <ruleBacktrackFailure()>
 <(nvaExceptionWrapperMap.(k))({NoViableAltException nvae =
 new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);
<@noViableAltException()>
throw nvae;})>
<endif>
}
<endif>
>>

nvaExceptionWrapperMap ::= [
 "1":"wrapNvaExceptionForK1",
 "2":"wrapNvaExceptionForK2",
 default:"wrapNvaExceptionForKN"
]

wrapNvaExceptionForK1(exceptionCode)
 ::= <<
<exceptionCode>
>>

wrapNvaExceptionForK2(exceptionCode) ::= <<
int nvaeMark = input.mark();
try {
 input.consume();
 <exceptionCode>
} finally {
 input.rewind(nvaeMark);
}
>>

wrapNvaExceptionForKN(exceptionCode) ::= <<
int nvaeMark = input.mark();
try {
 for (int nvaeConsume = 0; nvaeConsume \< <k> - 1; nvaeConsume++) {
  input.consume();
 }
 <exceptionCode>
} finally {
 input.rewind(nvaeMark);
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<edges; separator="\nelse ">
>>

/** A DFA
 state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);
```

```
<edges; separator="\nelse ">
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
 alt<decisionNumber>=<eotPredictsAlt>;
}
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr,
 targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
 <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
 alt<decisionNumber>=<eotPredictsAlt>;
<else>
 <ruleBacktrackFailure()>
 <(nvaExceptionWrapperMap.(k))({NoViableAltException nvae =
 new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);
<@noViableAltException()>
throw nvae;})>
<endif>
}
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
```

```
switch ( input.LA(<k>) ) {
 <edges; separator="\n">
}
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>)
 ) {
<edges; separator="\n">
<if(eotPredictsAlt)>
default:
 alt<decisionNumber>=<eotPredictsAlt>;
 break;
<endif>
}
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it |case <it>:}; separator="\n">
 {
 <targetState>
 }
 break;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input);
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa)
 ::= <<
static final String DFA<dfa.decisionNumber>_eotS =
 "<dfa.javaCompressedEOT; wrap="\"+\n\t\"">";
static final String DFA<dfa.decisionNumber>_eofS =
 "<dfa.javaCompressedEOF; wrap="\"+\n\t\"">";
static final String DFA<dfa.decisionNumber>_minS =
 "<dfa.javaCompressedMin; wrap="\"+\n\t\"">";
static final String DFA<dfa.decisionNumber>_maxS =
 "<dfa.javaCompressedMax; wrap="\"+\n\t\"">";
static final String DFA<dfa.decisionNumber>_acceptS =
 "<dfa.javaCompressedAccept; wrap="\"+\n\t\"">";
static final String DFA<dfa.decisionNumber>_specialS =
 "<dfa.javaCompressedSpecial; wrap="\"+\n\t\"">}>";
static final String[] DFA<dfa.decisionNumber>_transitionS = {
  <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
};

static final short[] DFA<dfa.decisionNumber>_eot =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eotS);
static final short[] DFA<dfa.decisionNumber>_eof =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eofS);
static final char[] DFA<dfa.decisionNumber>_min
 = DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
static final char[] DFA<dfa.decisionNumber>_max =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
static final short[] DFA<dfa.decisionNumber>_accept =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
static final short[] DFA<dfa.decisionNumber>_special =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_specialS);
static final short[][] DFA<dfa.decisionNumber>_transition;

static {
 int numStates = DFA<dfa.decisionNumber>_transitionS.length;
 DFA<dfa.decisionNumber>_transition = new short[numStates][];
 for (int i=0; i\<numStates; i++) {
  DFA<dfa.decisionNumber>_transition[i] = DFA.unpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
 }
}

protected class DFA<dfa.decisionNumber> extends DFA {

 public DFA<dfa.decisionNumber>(BaseRecognizer recognizer) {
  this.recognizer = recognizer;
  this.decisionNumber = <dfa.decisionNumber>;
  this.eot = DFA<dfa.decisionNumber>_eot;
  this.eof
 = DFA<dfa.decisionNumber>_eof;
  this.min = DFA<dfa.decisionNumber>_min;
  this.max = DFA<dfa.decisionNumber>_max;
  this.accept = DFA<dfa.decisionNumber>_accept;
  this.special = DFA<dfa.decisionNumber>_special;
  this.transition = DFA<dfa.decisionNumber>_transition;
 }
 @Override
 public String getDescription() {
  return "<dfa.description>";
 }
 <@errorMethod()>
<if(dfa.specialStateSTs)>
 @Override
 public int specialStateTransition(int s, IntStream _input) throws NoViableAltException {
  <if(LEXER)>
  IntStream input = _input;
  <endif>
  <if(PARSER)>
  TokenStream input = (TokenStream)_input;
  <endif>
  <if(TREE_PARSER)>
  TreeNodeStream input = (TreeNodeStream)_input;
  <endif>
  int _s = s;
  switch ( s ) {
  <dfa.specialStateSTs:{state |
  case <i0> : <! compressed special state numbers 0..n-1 !>
   <state>}; separator="\n">
  }
<if(backtracking)>
  if (state.backtracking>0) {state.failed=true; return -1;}
<endif>
  NoViableAltException nvae =
   new NoViableAltException(getDescription(),
 <dfa.decisionNumber>, _s, input);
  error(nvae);
  throw nvae;
 }
<endif>
}
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(1);
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
int index<decisionNumber>_<stateNumber> = input.index();
input.rewind();
<endif>
s = -1;
<edges; separator="\nelse ">
```

```
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}
>>

/** An edge pointing
 at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(pred,{})>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<stateNumber> >= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt)
```

```
 ::= "(input.LA(<k>) >= <lower> && input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= <<
<ranges; separator="||">
>>

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
 <scope.attributes:{it |<it.decl>;}; separator="\n">
}
protected Stack\<<scope.name>_scope> <scope.name>_stack = new Stack\<<scope.name>_scope>();
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
protected static class <scope.name>_scope {
 <scope.attributes:{it |<it.decl>;}; separator="\n">
}
protected Stack\<<scope.name>_scope> <scope.name>_stack = new Stack\<<scope.name>_scope>();
<endif>
<endif>
>>

returnStructName(r) ::= "<r.name>_return"

returnType(ruleDescriptor) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<elseif(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
%>

/** Generate
 the Java type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<returnScopeBaseType()>
<elseif(referencedRule.hasSingleReturnValue)>
```

```
<referencedRule.singleValueReturnType>
<else>
void
<endif>
%>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<javaTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;
%>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public
 static class <ruleDescriptor:returnStructName()> extends <returnScopeBaseType()> {
 <if(scope)><scope.attributes:{it |public <it.decl>;}; separator="\n"><endif>
 <@ruleReturnMembers()>
};
<endif>
>>

returnScopeBaseType() ::= <%
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope
%>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"
```

```
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1).<attr.name>
<else>
<if(index)>
<scope>_stack.elementAt(<index>).<attr.name>
<else>
<scope>_stack.peek().<attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1).<attr.name> =<expr>;
<else>
<if(index)>
<scope>_stack.elementAt(<index>).<attr.name> =<expr>;
<else>
<scope>_stack.peek().<attr.name>
 =<expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?((<returnType(referencedRule)>)<scope>).<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
%>
```

```
returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>;
<else>
<attr.name> =<expr>;
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/**
 ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.getText():null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.getType():0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.getLine():0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.getCharPositionInLine():0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.getChannel():0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.getTokenIndex():0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?Integer.valueOf(<scope>.getText()):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=null?(<castToLabelType({<scope>.start})>):null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=null?(<castToLabelType({<scope>.stop})>):null)"
ruleLabelPropertyRef_tree(scope,attr)
 ::= "(<scope>!=null?((<ASTLabelType>)<scope>.getTree()):null)"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
(<scope>!=null?(input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
 input.getTreeAdaptor().getTokenStopIndex(<scope>.start))):null)
<else>
(<scope>!=null?input.toString(<scope>.start,<scope>.stop):null)
<endif>
%>

ruleLabelPropertyRef_st(scope,attr) ::= "(<scope>!=null?((StringTemplate)<scope>.getTemplate()):null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
 "(<scope>!=null?<scope>.getType():0)"
```

```
lexerRuleLabelPropertyRef_line(scope,attr) ::=
 "(<scope>!=null?<scope>.getLine():0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
 "(<scope>!=null?<scope>.getCharPositionInLine():-1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
 "(<scope>!=null?<scope>.getChannel():0)"
lexerRuleLabelPropertyRef_index(scope,attr)
 ::=
 "(<scope>!=null?<scope>.getTokenIndex():0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
 "(<scope>!=null?<scope>.getText():null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
 "(<scope>!=null?Integer.valueOf(<scope>.getText()):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "(<castToLabelType(\"retval.start\")>)"
rulePropertyRef_stop(scope,attr) ::= "(<castToLabelType(\"retval.stop\")>)"
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(retval.start),
 input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
%>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr)
 ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "Integer.valueOf(<scope>.getText())"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <%
<if(backtracking)>
```

```
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
<action>
<endif>
%>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name,
 words64) ::= <<
public static final BitSet <name> = new BitSet(new long[]{<words64:{it |<it>L};separator=",">});
>>

codeFileExtension() ::= ".java"

true_value() ::= "true"
false_value() ::= "false"

isEmptyString ::= [
 "" : true,
 default : false
]

isTrue ::= [
 "true" : true,
 default : false
]

isDefaultLabelType ::= [
 "Token" : true,
 default : false
]

isPredefinedRewriteRule ::= [
 "topdown" : true,
 "bottomup" : true,
 default : false
]

castToLabelType(value) ::= <%
<if(!isDefaultLabelType.(labelType))>
(<labelType>)
<endif>
```

```
<value>
%>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/Java.stg
No license file was found, but licenses were detected in source scan.

```
group AST;

@outputFile.imports() ::= <<
<@super.imports()><if(!TREE_PARSER)><! tree parser would already have imported !>
Antlr.Runtime.Tree,<\n><endif>
>>

@genericParser.members() ::= <<
<@super.members()>
```

```
<parserMembers()>
>>

@genericParser.membersConstructor() ::= <<
<@super.membersConstructor()>
<parserMembersConstructor()>
>>

@genericParser.membersImplementation()
 ::= <<
<@super.membersImplementation()>
<parserMembersImplementation()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
 strict protected
   FAdaptor: ITreeAdaptor;
   procedure SetAdaptor(const Value: ITreeAdaptor);
   property Adaptor: ITreeAdaptor read FAdaptor;
 public
   property TreeAdaptor: ITreeAdaptor read FAdaptor write SetAdaptor;

>>

parserMembersConstructor() ::= <<
FAdaptor := TCommonTreeAdaptor.Create;
>>

parserMembersImplementation() ::= <<
procedure T<grammar.recognizerName>.SetAdaptor(const Value: ITreeAdaptor);
begin
 FAdaptor := Value;
 <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor := FAdaptor;}>
end;
>>

@returnScope.ruleReturnMembers() ::= <<
function T<grammar.recognizerName>.T<ruleDescriptor:returnStructName()>.GetTree: IANTLRInterface;
begin
 Result := FTree;
end;

procedure T<grammar.recognizerName>.T<ruleDescriptor:returnStructName()>.SetTree(const Value:
IANTLRInterface);
begin
 FTree := Value
 as I<ASTLabelType>;
end;
>>

@returnScopeDeclaration.ruleReturnMembers() ::= <<
strict private
 FTree: I<ASTLabelType>;
protected
 { IRuleReturnScope }
 function GetTree: IANTLRInterface; override;
 procedure SetTree(const Value: IANTLRInterface); override;
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
Root[0] := nil;<\n>
>>

ruleDeclarationVars() ::= <<
<super.ruleDeclarationVars()>
Root: array [0..63] of I<ASTLabelType>;
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{<it.label.text>_tree := nil;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<it.label.text>_tree := nil;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites:{Locals['Stream_<it>'] :=
TRewriteRule<rewriteElementType>Stream.Create(Adaptor,'token <it>');}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites:{Locals['Stream_<it>'] :=
TRewriteRuleSubtreeStream.Create(Adaptor,'rule <it>');}; separator="\n">
>>

ruleLabelDefVars()
 ::= <<
<super.ruleLabelDefVars()>
<ruleDescriptor.tokenLabels:{<it.label.text>_tree: I<ASTLabelType>;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<it.label.text>_tree: I<ASTLabelType>;}; separator="\n">
>>
/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
```

```
<if(!rewriteMode)>
Root[0] := Adaptor.GetNilNode as I<ASTLabelType>;
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<token>'] as
IRewriteRuleElementStream).Add(<label>);<\n>
>>

/** ids+=ID and track
 it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<token>'] as
IRewriteRuleElementStream).Add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<rule.name>'] as
IRewriteRuleElementStream).Add(<label>.Tree);<\n>
>>
```

```
/**
 x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<rule>'] as
IRewriteRuleElementStream).Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

// Rewrite

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST
 REWRITE
// elements:          <referencedElementsDeep; separator=", ">
// token labels:      <referencedTokenLabels; separator=", ">
// rule labels:       <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels:  <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
<prevRuleRootRef()>.Tree := Root[0];
<rewriteCodeLabels()>
```

```
Root[0] := Adaptor.GetNilNode as I<ASTLabelType>;
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.Tree = (<ASTLabelType>)adaptor.rulePostProcessing(root[0]);
input.ReplaceChildren(adaptor.GetParent(retval.Start),
                      adaptor.GetChildIndex(retval.Start),
                      adaptor.GetChildIndex(_last),
                      retval.Tree);
<endif>
<endif>
<! if parser or rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.Tree
 := Root[0];<\n>
<endif>
<if(!rewriteMode)>
<prevRuleRootRef()>.Tree := Root[0];<\n>
<endif>
<if(backtracking)>
end;
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{Locals['Stream_<it>'] := TRewriteRule<rewriteElementType>Stream.Create(Adaptor, 'token <it>', <it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{Locals['Stream_<it>'] := TRewriteRule<rewriteElementType>Stream.Create(Adaptor,'token <it>', list_<it>);};
   separator="\n"
>
<referencedRuleLabels:{
if Assigned(<it>) then
 Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', <it>.Tree)
else
 Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', nil);}; separator="\n">
<referencedRuleListLabels
   :{Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow
 like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
(* <fileName>:<description> *)
if (<referencedElementsDeep:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator="
or ">) then
begin
 <alt>
end;
<referencedElementsDeep:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset;<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
(* <fileName>:<description> *)
while (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">) do
begin
 <alt>
end;
<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep,
 // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
if (not (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">)) then
 raise ERewriteEarlyExitException.Create('');

while (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">) do
begin
 <alt>
end;
```

```
<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset();<\n>}>
>>

rewriteAlt(a) ::= <<
(* <a.description> *)
<if(a.pred)>
if (<a.pred>) then
begin
 <a.alt>
end<\n>
<else>
begin
 <a.alt>
end;<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "Root[0] = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
(* <fileName>:<description> *)
begin
 Root[<treeLevel>] := Adaptor.GetNilNode as I<ASTLabelType>;
 <root:rewriteElement()>
 <children:rewriteElement()>
 Adaptor.AddChild(Root[<enclosingTreeLevel>],
 Root[<treeLevel>]);
end;<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as
IRewriteRuleElementStream).NextNode());<\n>
>>
```

```
/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode);<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as
IRewriteRuleElementStream).NextNode(), Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/**
 Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>, Root[<treeLevel>]) as
I<ASTLabelType>;<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>, Root[<treeLevel>]) as
I<ASTLabelType>;<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
Root[0] = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the
 value of Root[0] right
*  before I set it during rewrites.  The assign will be to retval.Tree.
*/
prevRuleRootRef() ::= "RetVal"

rewriteRuleRef(rule) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<rule>'] as IRewriteRuleElementStream).NextTree());<\n>
>>
```

```
rewriteRuleRefRoot(rule) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<rule>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

rewriteNodeAction(action) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<action>, Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
Adaptor.AddChild(Root[<treeLevel>],
 (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
T<terminalOptions.node>.Create(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
Adaptor.CreateNode(<tokenType>, <args; separator=", "><if(!args)>'<tokenType>'<endif>) as I<ASTLabelType>
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
T<terminalOptions.node>.Create((Locals['Stream_<token>']
 as IRewriteRuleElementStream).NextToken<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
Adaptor.Create(<token>, <args; separator=", ">)
<else>
(Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Delphi/AST.stg
No license file was found, but licenses were detected in source scan.

```
dfa(decisionRanks,states,edges,rankdir,startState,useBox) ::= <<
digraph NFA {
<if(rankdir)>rankdir=<rankdir>;<endif>
<decisionRanks; separator="\n">
<states; separator="\n">
<edges; separator="\n">
}
>>

nfa(decisionRanks,states,edges,rankdir,startState) ::= <<
digraph NFA {
rankdir=LR;
<decisionRanks; separator="\n">
<states;
 separator="\n">
<edges; separator="\n">
}
>>

decision-rank(states) ::= <<
{rank=same; rankdir=TB; <states; separator="; ">}
>>

edge(src,target,label,arrowhead) ::= <<
<src> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7, label = "<label>"<if(arrowhead)>, arrowhead =
<arrowhead><endif>];
>>

action-edge(src,target,label,arrowhead) ::= <<
<src> -> <target> [fontsize=11, fontname="Courier", arrowsize=.7, label = "<label>"<if(arrowhead)>, arrowhead =
<arrowhead><endif>];
>>

epsilon-edge(src,target,label,arrowhead) ::= <<
<src> -> <target> [fontname="Times-Italic", label = "e"];
>>

state(name,useBox) ::= <<
node [fontsize=11, shape = <if(useBox)>box<else>circle, fixedsize=true, width=.4<endif>]; <name>
>>

stopstate(name,useBox) ::= <<
node [fontsize=11, shape = <if(useBox)>polygon,sides=4,peripheries=2<else>doublecircle, fixedsize=true,
width=.6<endif>]; <name>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/templates/dot/dot.stg

No license file was found, but licenses were detected in source scan.

```
/*
[The "BSD license"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports()>
import org.stringtemplate.v4.*;
import java.util.HashMap;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
/* ST returnScope.ruleReturnMembers -- empty */
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturn.memvars() ::= <<
ST *st;
```

```
>>

/** Add this
 to each rule's return value struct */
@returnScope.ruleReturn.properties() ::= <<
@property (retain) ST *st;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturn.methodsDecl() ::= <<
- (ST *)getTemplate;
- (NSString *)toString;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturn.synthesize() ::= <<
@synthesize st;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturn.methods() ::= <<
- (ST *)getTemplate { return st; }
- (NSString *)toString { return st==nil?nil:[st render]; }
>>

@genericParser.members() ::= <<
<@super.members()>
STGroup *templateLib = [STGroup newSTGroup];

- (void)setTemplateLib:(STGroup *)aTemplateLib
{
   self.templateLib = aTemplateLib;
}

- (STGroup *)getTemplateLib
{
   return templateLib;
}
>>

@genericParserHeaderFile.memVars() ::= <<
<@super.memVars()>
/* ST genericParserHeaderFile.memVars -- empty now */
STGroup *templateLib; /* ST -- really a part of STAttrMap */
>>

@genericParserHeaderFile.properties()
```

```
 ::= <<
<@super.properties()>
/* ST genericParser.properties */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
>>

@genericParserHeaderFile.methodsDecl() ::= <<
<@super.methodsDecl()>
/* ST genericParser.methodsDecl */
- init;
- (STGroup *) getTemplateLib;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
@end
>>

@genericParser.synthesize() ::= <<
<@super.synthesize()>
/* ST genericParserImplementation.synthesize */
@synthesize templateLib;
>>

@genericParser.methods() ::= <<
<@super.methods()>
/* ST genericParser.methods */

- (STGroup *)getTemplateLib
{
   return templateLib;
}

- (void) setTemplateLib:(STGroup *)aTemplateLib
{
   templateLib = aTemplateLib;
}

>>

@genericParser.members() ::= <<
<@super.members()>
STGroup *templateLib = [STGroup newSTGroup];

- (STGroup *) getTemplateLib
{
 return templateLib;
}

- (void) setTemplateLib:(STGroup *) templateLib
{
 this.templateLib = templateLib;
}

/** x+=rule when
 output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, { [<label> getTemplate]; })>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
 <alts:rewriteTemplateAlt(); separator="else ">
 <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
   [([(TokenRewriteStream)input getTokenStream])
   replaceFromIndex:[[input getTreeAdaptor] getTokenStartIndex:retval.start]
   ToIndex:[[input getTreeAdaptor] getTokenStopIndex:retval.start]
   Text:[retval.st render]];
<else>
   [((TokenRewriteStream)input)
   replaceFromIndex:[((Token)retval.start) getTokenIndex]
   ToIndex:[input LT:-1] getTokenIndex]
   Text:[retval.st render]];
<endif>
>>

rewriteTemplateAlt(alt) ::= <<
// <alt.description>
<if(alt.pred)>
if
 (<alt.pred>) {
   retval.st = <alt.alt>;
}<\n>
<else>
{
   retval.st = <alt.alt>;
```

```
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
nil;
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args) ::= <%
<if(args)><args:{a | [}><endif>
[templateLib getInstanceOf:@"<name>"]
 <if(args)><args:{a | add:@"<a.name>" value:<a.value>]}><endif>;
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
<if(args)><args:{a | [}><endif>
[templateLib getInstanceOf:<expr>];
<if(args)><args:{a | add:@"<a.name>" value:<a.value>]}><endif>;
%>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <%
<if(args)><args:{a
 | [}><endif>
[ST newST:templateLib template:"<template>"]
 <if(args)><args:{a |  add:@"<a.name>" <a.value>]}><endif>;
%>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
[(<st>) setAttribute:@"<attrName>" value:<expr>];
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
[ST newST:templateLib template:<stringExpr>];
```

```
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ST4ObjC.stg
No license file was found, but licenses were detected in source scan.

```
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@returnScopeInterface.memVars() ::= <<
<@super.memVars()>
/* ST returnInterface.memVars */
ST *st;
>>

@returnScopeInterface.properties() ::= <<
<@super.properties()>
/* ST returnScope.properties */
```

```
@property (retain, getter=getST, setter=setST:) ST *st;
>>

@returnScopeInterface.methodsDecl() ::= <<
<@super.methodsDecl()>
/* ST AST returnScopeInterface.methodsDecl */
-
 (id) getTemplate;
- (NSString *) toString;
- (NSString *) description;
>>

@returnScopeInterface() ::= <<
/* ST returnScopeInterface */
@interface <returnScopeInterface.name> : ReturnScope {
<returnScopeInterface.memVars()>
}
<returnScopeInterface.properties()>

<returnScopeInterface.methods()>
@end
>>

@returnScopeImplementation.synthesize() ::= <<
<@super.synthesize()>
/* ST returnScope.synthesize */
@synthesize st;
>>

@returnScopeImplementation.methods() ::= <<
<@super.methods()>
/* ST returnScope.methods */
- (id) getTemplate { return st; }
- (NSString *) toString { return st==nil?nil:[st toString]; }
- (NSString *) description { [self toString]; }
>>

@returnScopeImplementation() ::= <<
@implementation <returnScopeImplementation.name>
<returnScopeImplementation.synthesize()>

<returnScopeImplementation.methods()>
@end
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
<@super.ruleReturnMembers()>
```

```
/* ST returnScope.ruleReturnMembers --
 empty */
>>

@genericParserHeaderFile.memVars() ::= <<
<@super.memVars()>
/* ST genericParserHeaderFile.memVars -- empty now */
STGroup *templateLib; /* ST -- really a part of STAttrMap */
>>

@genericParserHeaderFile.properties() ::= <<
<@super.properties()>
/* ST genericParser.properties */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
>>

@genericParserHeaderFile.methodsDecl() ::= <<
<@super.methodsDecl()>
/* ST genericParser.methodsDecl */
- init;
- (STGroup *) getTemplateLib;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
@end
>>

@genericParser.synthesize() ::= <<
<@super.synthesize()>
/* ST genericParserImplementation.synthesize */
@synthesize templateLib;
>>

@genericParser.methods() ::= <<
<@super.methods()>
/* ST genericParser.methods */

- (STGroup *)getTemplateLib
{
   return templateLib;
}

- (void) setTemplateLib:(STGroup *)aTemplateLib
{
   templateLib = aTemplateLib;
}

>>

@genericParser.members() ::= <<
```

```
<@super.members()>
STGroup
 *templateLib = [STGroup  newSTGroup:@"<name>Templates"];

- (STGroup *) getTemplateLib
{
 return templateLib;
}

- (void) setTemplateLib:(STGroup *) templateLib
{
 this.templateLib = templateLib;
}

/** allows convenient multi-value initialization:
*  "new STAttrMap().put(...).put(...)"
*/
/* REPLACE THIS STATIC CLASS
static class STAttrMap extends HashMap {
- (STAttrMap *) setObject:(id)aValue forKey:(NS*)String attrName
{
   [super setObject:value forKey:attrName];
   return self;
}
- (STAttrMap *) setObjectWithInt:(NSInteger)value forKey:(NSString *)attrName
{
   [super setObject:[NSNumber numberWithInteger:value] forKey:attrName];
   return self;
 }
}
*/
>>

@STAttrMap() ::= <<
/* -------- ST start STAttrMap ------------ */
<@STAttrMap.interface()>
<@STAttrMap.implementation()>
/* -------- ST end STAttrMap ------------ */
>>

@STAttrMap.interface() ::= <<
/* -------- ST start STAttrMap.interface ------------ */
@interface STAttrMap : HashMap {
/*    <@STAttrMap.memVars()>
 */
   STGroup *templateLib;
}
```

```
/*    <@STAttrMap.properties()> */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
/* <@STAttrMap.methodsDecl()> */
- (id) init;
- (STAttrMap *) setObject:(id)value forKey:(NSString *)attrName;
- (STAttrMap *) setObjectWithInt:(NSInteger)value forKey:(NSString *)attrName;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
- (STGroup *) getTemplateLib;
@end
/* -------- ST end STAttrMap.interface ------------ */
>>

@STAttrMap.implementation() ::= <<
/* -------- ST start STAttrMap.implementation ------------ */
/** allows convenient multi-value initialization:
*  "new STAttrMap().put(...).put(...)"
*/
@implementation STAttrMap
@synthesize templateLib;

<@STAttrMap.methods()>
@end
/* -------- ST end STAttrMap.implementation ------------ */
>>

@STAttrMap.memVars() ::= <<
/* -------- ST start STAttrMap.memVars ------------ */
   STGroup *templateLib;
/* -------- ST end STAttrMap.memVars ------------ */
>>

@STAttrMap.properties()
 ::= <<
/* -------- ST start STAttrMap.properties ------------ */
@property (retain, getter=getTemplateLib, setter=setTemplateLib:) STGroup *templateLib;
/* -------- ST end STAttrMap.properties ------------ */
>>

@STAttrMap.methodsDecl() ::= <<
/* -------- ST start STAttrMap.methodsDecl ------------ */
- (id) init;
- (STAttrMap *) setObject:(id)value forKey:(NSString *)attrName;
- (STAttrMap *) setObjectWithInt:(NSInteger)value forKey:(NSString *)attrName;
- (void) setTemplateLib:(STGroup *)aTemplateLib;
- (STGroup *) getTemplateLib;
/* -------- ST end STAttrMap.methodsDecl ------------ */
>>
```

```
@STAttrMap.methods() ::= <<
/* -------- ST start STAttrMap.methods ------------ */
- (id) init
{
   self = [super initWithLen:16];
   if ( self != nil ) {
       templateLib = [STGroup newSTGroup:"<name>Templates"];
   }
   return self;
}

- (STAttrMap *) setObject:(id)aValue forKey:(NSString *)aAttrName
{
   [super setObject:aValue forKey:aAttrName];
   return self;
}

-
 (STAttrMap *) setObjectWithInt:(NSInteger)aValue forKey:(NSString *)aAttrName
{
   [super setObject:[NSNumber numberWithInteger:aValue] forKey:aAttrName];
   return self;
}
- (void) setTemplateLib:(STGroup *)aTemplateLib
{
   templateLib = aTemplateLib;
}

- (STGroup *)getTemplateLib
{
   return templateLib;
}
/* -------- ST end STAttrMap.methods ------------ */
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ST ruleRefAndListLable */
<ruleRef(...)>
<listLabel(elem=[label getTemplate,...]>
>>

rewriteTemplate(alts) ::= <<
/* -------- ST start rewriteTemplate ------------ */
// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
   <alts:rewriteTemplateAlt(); separator="else ">
   <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
/* -------- ST end rewriteTemplate ------------ */
>>

replaceTextInLine()
 ::= <<
/* -------- ST start replaceTextInLine ------------ */
<if(TREE_PARSER)>
[[(TokenRewriteStream *)input getTokenStream]
   replaceFromIndex:[[input getTreeAdaptor] getTokenStartIndex:retval.start]
            ToIndex:[[input getTreeAdaptor] getTokenStopIndex:retval.start]
               Text:retval.st];
<else>
[(TokenRewriteStream *)input)
       replaceFromIndex:[((CommonToken *)retval.start) getTokenIndex]
                ToIndex:[[input LT:-1] getTokenIndex]
                   Text:retval.st];
<endif>
/* -------- ST end replaceTextInLine ------------ */
>>

rewriteTemplateAlt() ::= <<
/* -------- ST start rewriteTemplateAlt ------------ */
/* ST <it.description> */
<if(it.pred)>
if (<it.pred>) {
   retval.st = <it.alt>;
}<\n>
<else>
{
   retval.st = <it.alt>;
}<\n>
<endif>
/* -------- ST end rewriteTemplateAlt ------------ */
>>

rewriteEmptyTemplate(alts) ::= <<
nil;
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value
```

```
 of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args) ::= <<
/* -------- ST start rewriteExternalTemplate ------------ */
[templateLib getInstanceOf:@"<name>"
<if(args)>[[STAttrMap newSTAttrMap] <args:{a | setObject:<a.value> forKey:@"<a.name>"]}><endif>]
/* -------- ST end rewriteExternalTemplate ------------ */
>>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <<
/* -------- ST start rewriteIndirectTemplate ------------ */
[templateLib getInstanceOf:<expr>
<if(args)> [[STAttrMap newSTAttrMap]<args:{a | setObject:<a.value> forKey:@"<a.name>"]}>]
<else>]<endif>
/* -------- ST end rewriteIndirectTemplate ------------ */
>>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <<
/* -------- ST
 start rewriteInlineTemplate ------------ */
STGroup *templateLib;
templateLib.templates = [STAttrMap newSTAttrMap];
<if(args)> [templateLib.templates <args:{a | setObject:<a.value> forKey:@"<a.name>"];}><endif>
[ST newST:templateLib template:@"<template>"];
/* -------- ST end rewriteInlineTemplate ------------ */
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
/* -------- ST start rewriteAction ------------ */
<action>
/* -------- ST end rewriteAction ------------ */
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
/* -------- ST start actionSetAttribute ------------ */
[[ST attributes] setObject:<expr> forKey:@"<attrName>"];
<![<st> setAttribute:<expr> name:@"<attrName>"];!>
/* -------- ST end actionSetAttribute ------------ */
>>

/** Translate %{stringExpr} */
```

```
actionStringConstructor(stringExpr) ::= <<
/* -------- ST start actionStringConstructor ------------ */
[ST newSTWithTemplate:<stringExpr>]
/* --------
 ST end actionStringConstructor ------------ */
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ST.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
```

```
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/

/* addition memVars for returnscopes */
@returnScopeInterface.memVars() ::= <<
/* ASTTreeParser returnScopeInterface.memVars
 */
<recognizer.ASTLabelType; null="CommonTree"> *tree;
>>

/** the interface of returnScope methodsDecl */
@returnScopeInterface.methodsDecl() ::= <<
/* ASTTreeParser returnScopeInterface.methodsDecl */
- (<recognizer.ASTLabelType; null="CommonTree"> *)getTree;
- (void) setTree:(<recognizer.ASTLabelType; null="CommonTree"> *)aTree;<\n>
>>

/** the implementation of returnScope methods */
@returnScope.methods() ::= <<
/* ASTTreeParser returnScope.methods */
- (<ASTLabelType> *)getTree
{
   return tree;
}

- (void) setTree:(<ASTLabelType> *)aTree
{
   if (tree != aTree) {
       if (tree != nil) [tree release];
       if (aTree != nil) [aTree retain];
       tree = aTree;
   }
}

- (void) dealloc
{
   [self setTree:nil];
   [super dealloc];
}

@synthesize tree;
>>

@returnScopeProperties() ::= <<
@property (retain) <recognizer.ASTLabelType; null="CommonTree"> *tree;
>>
```

```
/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
/* ASTTreeParser
 ruleDeclarations */
<super.ruleDeclarations()>
<ASTLabelType> *_first_0 = nil;
<ASTLabelType> *_last = nil;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
/* ASTTreeParser noRewrite */
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType> *)_first_0;
if ( [treeAdaptor getParent:retval.tree] != nil && [treeAdaptor isNil:[treeAdaptor getParent:retval.tree]] ) )
   retval.tree = (<ASTLabelType> *)[treeAdaptor getParent:retval.tree];
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
/* ASTTreeParser tree */
_last = (<ASTLabelType> *)[input LT:1];
{
<ASTLabelType> *_save_last_<treeLevel> = _last;
<ASTLabelType> *_first_<treeLevel> = nil;
<if(!rewriteMode)>
<ASTLabelType>
 *root_<treeLevel> = [[[treeAdaptor class] newEmptyTree] retain];
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==nil ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==nil ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
```

```
<if(nullableChildList)>
if ( [input LA:1] == TokenTypeDOWN ) {
   [self match:input TokenType:TokenTypeDOWN Follow:nil]; <checkRuleBacktrackFailure()>
   <children:element()>
   [self match:input TokenType:TokenTypeUP Follow:nil]; <checkRuleBacktrackFailure()>
}
<else>
[self match:input TokenType:TokenTypeDOWN Follow:nil]; <checkRuleBacktrackFailure()>
<children:element()>
[self match:input TokenType:TokenTypeUP Follow:nil]; <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
[treeAdaptor addChild:root_<treeLevel> toTree:root_<enclosingTreeLevel>];
<endif>
_last
 = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRefBang */
_last = (<ASTLabelType> *)[input LT:1];
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRef */
_last = (<ASTLabelType> *)[input LT:1];
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
   <label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
   <label>_tree = (<ASTLabelType> *)[treeAdaptor dupNode:<label>];
<endif><\n>
   [treeAdaptor addChild:<label>_tree toTree:root_<treeLevel>];
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==nil
 ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRefAndListLabel */
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex) ::= <<
/* ASTTreeParser tokenRefRuleRoot */
_last = (<ASTLabelType> *)[input LT:1];
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
<label>_tree = (<ASTLabelType> *)[treeAdaptor dupNode:<label>];
<endif><\n>
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>_tree old:root_<treeLevel>];
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser tokenRefRuleRootAndListLabel
 */
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions) ::= <<
/* ASTTreeParser wildcard */
_last = (<ASTLabelType> *)[input LT:1];
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType> *)[adaptor dupTree:<label>];
[adaptor addChild:<label>_tree toTree:root_<treeLevel>];
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
```

```
if ( _first_<treeLevel> == nil ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
/* ASTTreeParser matchSet */
_last = (<ASTLabelType> *)[input LT:1];
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = [<terminalOptions.node>
 new<terminalOptions.node>:<label>];
<else>
<label>_tree = (<ASTLabelType> *)[adaptor dupNode:<label>];
<endif><\n>
[adaptor addChild:<label>_tree toTree:root_<treeLevel>];
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
/* ASTTreeParser matchRuleBlockSet */
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
>>

matchSetBang(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
/* ASTTreeParser matchSetBang */
_last = (<ASTLabelType> *)[input LT:1];
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
/* ASTTreeParser matchSetRuleRoot */
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(terminalOptions.node)>
<label>_tree = [<terminalOptions.node> new<terminalOptions.node>:<label>];
<else>
<label>_tree = (<ASTLabelType> *)[adaptor dupNode:<label>];
<endif><\n>
root_<treeLevel>
```

```
 = (<ASTLabelType> *)[adaptor becomeRoot:<label>_tree old:root_<treeLevel>];
<if(backtracking)>\}<endif>
<endif>
}, ...
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRef */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
   [treeAdaptor addChild:<label>.tree toTree:root_<treeLevel>];
<else> <! rewrite mode !>
if ( _first_<treeLevel> == nil ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefAndListLabel */
<ruleRef(...)>
<! <listLabel(elem = "["+label+" getTree]",...)> !>
<listLabel(elem = {[<label> getTree]},...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRoot */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if
 ( state.backtracking == 0 ) <endif>
root_<treeLevel> = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>.tree old:root_<treeLevel>];
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRootAndListLabel */
<ruleRefRuleRoot(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>
```

```
/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefTrack */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefTrackAndListLabel */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser
 ruleRefRuleRootTrack */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTTreeParser ruleRefRuleRootTrackAndListLabel */
_last = (<ASTLabelType> *)[input LT:1];
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
/* ASTTreeParser createRewriteNodeFromElement */
<if(terminalOptions.node)>
<! new <terminalOptions.node>(stream_<token>.nextNode()) !>
[[[<terminalOptions.node>(stream_<token> alloc] init] nextNode];
<else>
<! stream_<token>.nextNode() !>
[stream_<token> nextNode]
<endif>
>>

ruleCleanUp() ::= <<
/* ASTTreeParser ruleCleanUp */
<super.ruleCleanUp()>
<if(!rewriteMode)>
```

```
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = (<ASTLabelType> *)[treeAdaptor
 rulePostProcessing:root_0];
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
 */
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_First[0] := nil;
_Last := nil;<\n>
>>

ruleDeclarationVars() ::= <<
<super.ruleDeclarationVars()>
_First, _Save_Last: array [0..63] of I<ASTLabelType>;
_Last: I<ASTLabelType>;
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if (State.Backtracking = 0) then
begin<endif>
<if(rewriteMode)>
RetVal.Tree := _First[0] as I<ASTLabelType>;
if (Adaptor.GetParent(RetVal.Tree) \<\> nil) and (Adaptor.IsNil(Adaptor.GetParent(RetVal.Tree))) then
 RetVal.Tree := Adaptor.GetParent(RetVal.Tree) as I<ASTLabelType>;
<endif>
<if(backtracking)>end;<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_Last := Input.LT(1)
 as I<ASTLabelType>;
begin
 _Save_Last[<treeLevel>] := _Last;
 _First[<treeLevel>] := nil;
<if(!rewriteMode)>
 Root[<treeLevel>] := Adaptor.GetNilNode as I<ASTLabelType>;<\n>
```

```
<endif>
 <root:element()>
<if(rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
<if(root.el.rule)>
 if (_First[<enclosingTreeLevel>] = nil) then _First[<enclosingTreeLevel>] := <root.el.label>.Tree;
<else>
 if (_First[<enclosingTreeLevel>] = nil) then _First[<enclosingTreeLevel>] := <root.el.label>;
<endif>
<endif>
 <actionsAfterRoot:element()>
<if(nullableChildList)>
 if (Input.LA(1) = TToken.DOWN) then
 begin
   Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
 end;
<else>
 Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
 Adaptor.AddChild(Root[<enclosingTreeLevel>],
 Root[<treeLevel>]);
<endif>
 _Last := _Save_Last[<treeLevel>];
end;<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
```

```
begin<\n>
<endif>
<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
 <label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
 Adaptor.AddChild(Root[<treeLevel>], <label>_tree);
<if(backtracking)>
end;
<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
if (_First[<treeLevel>] = nil) then _First[<treeLevel>] := <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
 <label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
 Root[<treeLevel>] := Adaptor.BecomeRoot(<label>_tree, Root[<treeLevel>]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRoot(...)>
```

```
<listLabel(elem=label,...)>
>>

// SET AST

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <<
_Llast := Input.LT(1) as I<ASTLabelType>;
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if
 (State.Backtracking = 0) then begin <endif>
<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Adaptor.AddChild(Root[<treeLevel>], <label>_tree);
<if(backtracking)>end;<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then begin <endif>
<if(terminalOptions.node)>
<label>_tree := T<terminalOptions.node>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Root[<treeLevel>] := Adaptor.BecomeRoot(<label>_tree,
 Root[<treeLevel>]) as I<ASTLabelType>;
<if(backtracking)>end;<endif>
<endif>
}
)>
```

```
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
<if(!rewriteMode)>
Adaptor.AddChild(Root[<treeLevel>], <label>.Tree);
<else> <! rewrite mode !>
if (_First[<treeLevel>] = nil) then _First[<treeLevel>] := <label>.Tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[<treeLevel>] :=
Adaptor.BecomeRoot(<label>.Tree, Root[<treeLevel>]) as I<ASTLabelType>;
<endif>
>>

/**
 ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
```

```
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken
 to nextNode.
*/
createRewriteNodeFromElement(token,terminalOptions,scope) ::= <<
<if(terminalOptions.node)>
T<terminalOptions.node>.Create((Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode)
<else>
(Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
RetVal.Tree := Adaptor.RulePostProcessing(Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Delphi/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

```
/** Add an adaptor property that knows how to build trees */
@headerFile.members() ::= <<
/* @headerFile.members() */
pANTLR3_BASE_TREE_ADAPTOR adaptor;
pANTLR3_VECTOR_FACTORY  vectors;
/* End @headerFile.members() */
>>

/** Install the tree adaptor interface pointer and anything else that
*  tree parsers and producers require.
*/
@genericParser.apifuncs() ::= <<
<if(PARSER)>
ADAPTOR = ANTLR3_TREE_ADAPTORNew(instream->tstream->tokenSource->strFactory);<\n>
<endif>
ctx->vectors =
 antlr3VectorFactoryNew(0);
>>
```

```
@genericParser.cleanup() ::= <<
ctx->vectors->close(ctx->vectors);
<if(PARSER)>
/* We created the adaptor so we must free it
*/
ADAPTOR->free(ADAPTOR);
<endif>
>>

@returnScope.ruleReturnMembers() ::= <<

<super.ASTLabelType()> tree;

>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0;<\n>
>>

ruleInitializations() ::= <<
<super.ruleInitializations()>
root_0 = NULL;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{it | <ASTLabelType> <it.label.text>_tree;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <ASTLabelType> <it.label.text>_tree;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{it | pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{it | pANTLR3_REWRITE_RULE_SUBTREE_STREAM
 stream_<it>;}; separator="\n">
>>

/* Note that we defer the actual creation of any rewrite streams we need here and just initialize
* them to NULL. This saves creating huge numbers of rewrite streams that cannot be used as only
* one alt will be taken in a rule, but we are declaring all the streams here. So we define
* a macro that conatins the create code, then use this macro later to check if the stream
* has been created yet. Checking for NULL is almost free in C.
*/
ruleLabelInitializations() ::= <<
<super.ruleLabelInitializations()>
<ruleDescriptor.tokenLabels:{it | <it.label.text>_tree   = NULL;}; separator="\n">
<ruleDescriptor.tokenListLabels:{it | <it.label.text>_tree   = NULL;}; separator="\n">
```

```
<ruleDescriptor.allTokenRefsInAltsWithRewrites
:{it | stream_<it>   = NULL;
#define CREATE_stream_<it>  if (stream_<it> == NULL) {stream_<it> =
antlr3RewriteRule<rewriteElementType>StreamNewAE(ADAPTOR, RECOGNIZER, (pANTLR3_UINT8)"token
<it>"); \} };
 separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
:{it | stream_<it>   = NULL;
#define CREATE_stream_<it>  if (stream_<it> == NULL) {stream_<it> =
antlr3RewriteRuleSubtreeStreamNewAE(ADAPTOR, RECOGNIZER, (pANTLR3_UINT8)"rule <it>"); \}};
separator="\n">

<if(ruleDescriptor.hasMultipleReturnValues)>
retval.tree  = NULL;
<endif>
>>



/**  a rule label including default value */
ruleLabelInitVal(label) ::= <<
<super.ruleLabelInitVal(...)>
<label.label.text>.tree = NULL;
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<endif>
<endif>
>>

@alt.initializations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));<\n>
<endif>
<endif>
<endif>
>>


// Tracking Rule Eleme
```

```
 n t s
//
/** ID but track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<token>;
stream_<token>->add(stream_<token>, <label>, NULL);<if(backtracking)> }<endif><\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>CREATE_stream_<token>; stream_<token>-
>add(stream_<token>, <label>, NULL);<if(backtracking)> }<endif><\n>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/**
 rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<rule.name>;
stream_<rule.name>->add(stream_<rule.name>, <label>.tree, NULL);<if(backtracking)> }<endif>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelTrack(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<rule.name>;
stream_<rule.name>->add(stream_<rule.name>, <label>.tree, NULL);<if(backtracking)> }<endif>
>>
```

```
/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelAST(...)>
>>


// RULE REF AST




/** Match ^(label+=TOKEN
 ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>


/* How to accumulate lists when we are doing rewrite tracking...
*/
listLabelTrack(label) ::= <<
/* listLabelTrack(label)
*/
if (list_<label> == NULL)
{
   list_<label>=ctx->vectors->newVector(ctx->vectors);
}
list_<label>->add(list_<label>, <label>.tree, NULL);
>>

/* How to accumulate lists of rule outputs (only allowed with AST
* option but if the user is going to walk the tree, they will want
* all their custom elements from rule returns.
*
* Normally, we use inline structures (which the compiler lays down
* code to copy from heap allocations. However, here we want to accumulate copies
* of the returned structures because we are adding them to a list. This only makes sense if the
* grammar is not rewriting the tree as a tree rewrite only preserves the tree, not the object/structure
* returned from the
 rule. The rewrite will extract the tree pointer. However, if we are not going to
* do a tree re-write, then the user may wish to iterate the structures returned by the rule in
* action code and will expect the user defined returns[] elements to be available when they do this.
* Hence we cannot just preserve the tree that was returned. So, we must copy the local structure and provide
* a function that can free the allocated space. We cannot know how to free user allocated elements and
* presume that the user will know to do this using their own factories for the structures they allocate.
*/
```

```
listLabelAST(label) ::= <<
if (list_<label> == NULL)
{
   list_<label>=ctx->vectors->newVector(ctx->vectors);
}
{
   RETURN_TYPE_<label> * tcopy;

   tcopy = (RETURN_TYPE_<label> *)ANTLR3_MALLOC(sizeof(RETURN_TYPE_<label>)); /* Note no
memory allocation checks! */
   ANTLR3_MEMCPY((void *)(tcopy), (const void *)&<label>, sizeof(RETURN_TYPE_<label>));
   list_<label>->add(list_<label>, (void
 *)tcopy, freeScope);  /* Add whatever the return type is */<\n>
}
>>

// R e w r i t e

rewriteCode(
 alts,
 description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel,
 enclosingTreeLevel,
 treeLevel) ::=
<<

/* AST REWRITE
* elements          : <referencedElementsDeep; separator=", ">
* token labels      : <referencedTokenLabels; separator=", ">
* rule labels       : <referencedRuleLabels; separator=", ">
* token list labels : <referencedTokenListLabels; separator=", ">
* rule list labels  : <referencedRuleListLabels; separator=", ">
*/
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) <\n>
<endif>
{
 <rewriteCodeLabelsDecl()>
 <rewriteCodeLabelsInit()>
 root_0       = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
 <prevRuleRootRef()>.tree    = root_0;
 <alts:rewriteAlt();
 separator="else ">
 <if(TREE_PARSER)>
 <if(rewriteMode)>
 <prevRuleRootRef()>.tree = (<ASTLabelType>)(ADAPTOR->rulePostProcessing(ADAPTOR, root_0));
 INPUT->replaceChildren(INPUT, ADAPTOR->getParent(ADAPTOR, retval.start),
                    ADAPTOR->getChildIndex(ADAPTOR, retval.start),
                    ADAPTOR->getChildIndex(ADAPTOR, _last),
                    retval.tree);
 <endif>
 <endif>
 <prevRuleRootRef()>.tree = root_0; // set result root
 <rewriteCodeLabelsFree()>

}
>>

rewriteCodeLabelsDecl() ::= <<
<referencedTokenLabels
  :{it | pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;};
  separator="\n"
>
<referencedTokenListLabels
  :{it | pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;};
  separator="\n"
>
<referencedRuleLabels
  :{it | pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;};
  separator="\n"
>
<referencedRuleListLabels
  :{it | pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;};
  separator="\n"
>
>>

rewriteCodeLabelsInit()
 ::= <<
<referencedTokenLabels
:{it | stream_<it>=antlr3RewriteRule<rewriteElementType>StreamNewAEE(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", <it>);};
separator="\n"
>
<referencedTokenListLabels
:{it | stream_<it>=antlr3RewriteRule<rewriteElementType>StreamNewAEV(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", list_<it>); };
separator="\n"
>
```

```
<referencedRuleLabels
:{it | stream_<it>=antlr3RewriteRuleSubtreeStreamNewAEE(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", <it>.tree != NULL ? <it>.tree : NULL);};
separator="\n"
>
<referencedRuleListLabels
:{it | stream_<it>=antlr3RewriteRuleSubtreeStreamNewAEV(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", list_<it>);};
separator="\n"
>
>>
rewriteCodeLabelsFree() ::= <<
<referencedTokenLabels
:{it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>); };
separator="\n"
>
<referencedTokenListLabels
:{it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
<referencedRuleLabels
:{it
 | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
<referencedRuleListLabels
:{it | if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,
 rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,  // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
{
 if ( <referencedElementsDeep:{el | (stream_<el> != NULL && stream_<el>->hasNext(stream_<el>)) };
separator="|| "> )
 {
  <alt>
 }
 <referencedElementsDeep:{el | if ( stream_<el> != NULL) stream_<el>->reset(stream_<el>);<\n>}>
}<\n>
>>

rewriteClosureBlock(
 alt,
 rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,  // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
{
 while
 ( <referencedElements:{el | (stream_<el> != NULL && stream_<el>->hasNext(stream_<el>)) }; separator="|| "> )
 {
  <alt>
 }
 <referencedElements:{el | if (stream_<el> != NULL) stream_<el>->reset(stream_<el>);<\n>}>
}<\n>
>>
RewriteEarlyExitException() ::=
<<
CONSTRUCTEX();
EXCEPTION->type         = ANTLR3_REWRITE_EARLY_EXCEPTION;
EXCEPTION->name         = (void *)ANTLR3_REWRITE_EARLY_EXCEPTION_NAME;
>>
rewritePositiveClosureBlock(
 alt,
 rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,  // elements in immediately block; no nested blocks
 description) ::=
<<
if (<referencedElements:{el | (stream_<el> == NULL || !stream_<el>->hasNext(stream_<el>)) }; separator="|| "> )
{
   <RewriteEarlyExitException()>
}
else
{
 while ( <referencedElements:{el | (stream_<el>->hasNext(stream_<el>)) }; separator="|| "> ) {
  <alt>
 }
 <referencedElements:{el | stream_<el>->reset(stream_<el>);<\n>}>
}
>>

rewriteAlt(a) ::= <<
```

```
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
    <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = NULL; /* \<-- rewriteEmptyAlt()) */"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
   <ASTLabelType> root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
   <root:rewriteElement()>
   <children:rewriteElement()>
   ADAPTOR->addChild(ADAPTOR, root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,terminalOptions,args) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextNode(stream_<label>));<\n>
>>

/**
 Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextNode(stream_<label>));<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRootToken(ADAPTOR, stream_<label> == NULL ?
NULL : stream_<label>->nextToken(stream_<label>), root_<treeLevel>));<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,terminalOptions,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR,
<createRewriteNodeFromElement(...)>, root_<treeLevel>));<\n>
>>

rewriteImaginaryTokenRef(args,token,terminalOptions,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <createImaginaryNode(tokenType=token,
 ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,terminalOptions,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR,
<createImaginaryNode(tokenType=token, ...)>, root_<treeLevel>));<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule,dup) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<rule> == NULL ? NULL : stream_<rule>-
>nextTree(stream_<rule>));<\n>
>>

rewriteRuleRefRoot(rule,dup) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, stream_<rule> == NULL ? NULL
: stream_<rule>->nextNode(stream_<rule>), root_<treeLevel>));<\n>
```

```
>>

rewriteNodeAction(action)
 ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <action>, root_<treeLevel>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, stream_<label> == NULL ? NULL
: stream_<label>->nextNode(stream_<label>), root_<treeLevel>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label)
 ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot((<ASTLabelType>)(stream_<label> == NULL
? NULL : stream_<label>->nextNode(stream_<label>), root_<treeLevel>));<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

createImaginaryNode(tokenType,terminalOptions,args) ::= <<
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
<terminalOptions.node>New(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
<if(args)>
```

```
#if <length(args)> == 2
 (<ASTLabelType>)ADAPTOR->createTypeTokenText(ADAPTOR, <tokenType>, TOKTEXT(<args; separator=", ">))
#else
 (<ASTLabelType>)ADAPTOR->createTypeText(ADAPTOR, <tokenType>, (pANTLR3_UINT8)<args; separator=", ">)
#endif

<else>
(<ASTLabelType>)ADAPTOR->createTypeText(ADAPTOR, <tokenType>, (pANTLR3_UINT8)"<tokenType>")
<endif>
<endif>
>>

createRewriteNodeFromElement(token,terminalOptions,args)
 ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>New(stream_<token>->nextToken(stream_<token>)<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>

#if <length(args)> == 2
ADAPTOR->createTypeTokenText(ADAPTOR, <token>->getType(<token>, TOKTEXT(<token>, <args; separator=", ">)) /* JIMI */
#else
ADAPTOR->createTypeToken(ADAPTOR, <token>->getType(<token>, <token>, <args; separator=", ">)
#endif

<else>
stream_<token> == NULL ? NULL : stream_<token>->nextNode(stream_<token>)
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/C/AST.stg

No license file was found, but licenses were detected in source scan.

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/

asTypeInitMap ::= [
 "int":"0",
 "uint":"0",
 "Number":"0.0",
 "Boolean":"false",
 default:"null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds,
 memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
package<if(actions.(actionScope).package)> <actions.(actionScope).package><endif> {
   <actions.(actionScope).header>
   <@imports>
import org.antlr.runtime.*;
<if(TREE_PARSER)>
   import org.antlr.runtime.tree.*;
<endif>
```

```
   <@end>

   <docComment>
   <recognizer>
}
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="Token",
     superClass="Lexer") ::= <<
public class <grammar.recognizerName> extends
<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else><@superClassName><superClass
><@end><endif> {
   <tokens:{it |public static const <it.name>:int=<it.type>;}; separator="\n">
   <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>
   <actions.lexer.members>

   // delegates
   <grammar.delegates:
        {g|public var <g:delegateName()>:<g.recognizerName>;};
 separator="\n">
   // delegators
   <grammar.delegators:
        {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   <last(grammar.delegators):{g|public var gParent:<g.recognizerName>;}>

   public function <grammar.recognizerName>(<grammar.delegators:{g|<g:delegateName()>:<g.recognizerName>,
}>input:CharStream = null, state:RecognizerSharedState = null) {
       super(input, state);
       <cyclicDFAs:cyclicDFACtor()>
<if(memoize)>
<if(grammar.grammarIsRoot)>
       this.state.ruleMemo = new Array(<numRules>+1);<\n> <! index from 1..n !>
<endif>
<endif>
       <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>this,
input, this.state);}; separator="\n">
       <grammar.delegators:
        {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
       <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
   }
   public override
 function get grammarFileName():String { return "<fileName>"; }

<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">
   <synpreds:{p | <lexerSynpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public override function nextToken():Token {
   while (true) {
       if ( input.LA(1)==CharStreamConstants.EOF ) {
           return TokenConstants.EOF_TOKEN;
       }
       this.state.token = null;
     this.state.channel = TokenConstants.DEFAULT_CHANNEL;
       this.state.tokenStartCharIndex = input.index;
       this.state.tokenStartCharPositionInLine = input.charPositionInLine;
        this.state.tokenStartLine = input.line;
     this.state.text = null;
       try {
           var m:int = input.mark();
           this.state.backtracking=1; <! means we won't throw slow exception !>
           this.state.failed=false;
           mTokens();
           this.state.backtracking=0;
           <! mTokens backtracks with synpred at backtracking==2
              and we set the synpredgate to allow actions at level 1. !>
           if ( this.state.failed ) {
               input.rewindTo(m);
               input.consume(); <! advance one char and try again !>
           }
           else {
               emit();
               return this.state.token;
           }
       }
       catch (re:RecognitionException) {
           // shouldn't happen in backtracking mode, but...
           reportError(re);
           recover(re);
       }
   }
   // Not reached - For ActionScript compiler
   throw new Error();
}

public override function memoize(input:IntStream,
  ruleIndex:int,
  ruleStartIndex:int):void
{
if
 ( this.state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

public override function alreadyParsedRule(input:IntStream, ruleIndex:int):Boolean {
if ( this.state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>

actionGate() ::= "this.state.backtracking==0"

filteringActionGate() ::= "this.state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
            labelType, members, rewriteElementType,
            filterMode, ASTLabelType="Object") ::= <<
public class <grammar.recognizerName> extends
<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else><@superClassName><superClass
><@end><endif> {
<if(grammar.grammarIsRoot)>
   public static const tokenNames:Array = [
       "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>",
 <tokenNames; separator=", ">
   ];<\n>
<endif>
   <tokens:{it |public static const <it.name>:int=<it.type>;}; separator="\n">

   // delegates
   <grammar.delegates: {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   // delegators
   <grammar.delegators:
       {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   <last(grammar.delegators):{g|public var gParent:<g.recognizerName>;}>

   <scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>
   <@members>
  <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
   public function <grammar.recognizerName>(<grammar.delegators:{g|<g:delegateName()>:<g.recognizerName>,
}>input:<inputStreamType>, state:RecognizerSharedState = null) {
       super(input, state);
       <cyclicDFAs:cyclicDFACtor()>
       <parserCtorBody()>
       <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>,
 }>this, input, this.state);}; separator="\n">
       <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
       <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
   }
   <@end>

   public override function get tokenNames():Array { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; }
   public override function get grammarFileName():String { return "<fileName>"; }

   <members>

   <rules; separator="\n\n">

   <! generate rule/method definitions for imported rules so they
      appear to be defined in this recognizer. !>
      // Delegated rules
   <grammar.delegatedRules:{ruleDescriptor|
       public function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>):<returnType()> {
<if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>}; separator=", ">);
 \}}; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:{it | <bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                        words64=it.bits)>}>
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new Array(<length(grammar.allImportedRules)>+1);<\n> <! index from 1..n !>
<endif>
<endif>
```

```
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType="Object",
superClass="Parser", labelType="Token", members={<actions.parser.members>}) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, "TokenStream", superClass,
            labelType, members, "Token",
            false, ASTLabelType)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different
 type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
          numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="Object",
          superClass="TreeParser", members={<actions.treeparser.members>}) ::= <<
<genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, "TreeNodeStream", superClass,
            labelType, members, "Node",
            filterMode, ASTLabelType)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public
 final function <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>):void {
   <ruleLabelDefs()>
<if(trace)>
   traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
   try {
       <block>
   }
   finally {
       traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
   }
<else>
   <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public final function <name>():Boolean {
   this.state.backtracking++;
   <@start()>
   var start:int = input.mark();
   try {
       <name>_fragment(); // can never throw exception
   } catch (re:RecognitionException) {
       trace("impossible: "+re);
   }
   var success:Boolean = !this.state.failed;
   input.rewindTo(start);
   <@stop()>
   this.state.backtracking--;
   this.state.failed=false;
   return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( this.state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) )
 { return <ruleReturnValue()>; }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (this.state.failed) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (this.state.backtracking>0) {this.state.failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
```

```
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
// $ANTLR start <ruleName>
// <fileName>:<description>
public final function <ruleName>(<ruleDescriptor.parameterScope:parameterScope()>):<returnType()> {
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
    <@preamble()>
   try {
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
   }
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   catch (re:RecognitionException) {
       reportError(re);
       recoverStream(input,re);
       <@setErrorReturnValue()>
   }<\n>
<endif>
<endif>
<endif>
   finally {
       <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
       <memoize()>
       <ruleScopeCleanUp()>
       <finally>
   }
   <@postamble()>
   return <ruleReturnValue()>;
}
// $ANTLR end <ruleName>
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
   <e.action>
}
```

```
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
var retval:<returnType()> = new <returnType()>();
retval.start = input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{
 a |
var <a.name>:<a.type> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
var <ruleDescriptor.name>_StartIndex:int = input.index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.push(new Object());}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.push(new Object());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it |<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{it |<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it |var <it.label.text>:<labelType>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it |var list_<it.label.text>:Array=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef();
 separator="\n">
<ruleDescriptor.ruleListLabels:{ll|var <ll.label.text>:RuleReturnScope = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |var <it.label.text>:<labelType>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |var <it.label.text>:int;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{it |var list_<it.label.text>:Array=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( this.state.backtracking>0 ) { memoize(input,
 <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start <ruleName>
public final function m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>):void {
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   try {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
```

```
        <block><\n>
<else>
        var _type:int = <ruleName>;
        var _channel:int = DEFAULT_TOKEN_CHANNEL;
        <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
        <ruleDescriptor.actions.init>
        <block>
        <ruleCleanUp()>
        this.state.type = _type;
        this.state.channel = _channel;
        <(ruleDescriptor.actions.after):execAction()>
<endif>
     }
    finally {
        <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
        <ruleScopeCleanUp()>
        <memoize()>
    }
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
public override function mTokens():void {
    <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber>:int=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
    <alts:{a | <altSwitchCase(i, a)>}>
}
<@postbranch()>
>>
```

```
/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var
 alt<decisionNumber>:int=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
   <alts:{a | <altSwitchCase(i, a)>}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
var cnt<decisionNumber>:int=0;
<decls>
<@preloop()>
loop<decisionNumber>:
do {
   var alt<decisionNumber>:int=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
 {
 <alts:{a | <altSwitchCase(i, a)>}>
 default :
     if ( cnt<decisionNumber> >= 1 ) break loop<decisionNumber>;
     <ruleBacktrackFailure()>
           throw new EarlyExitException(<decisionNumber>, input);
           <! Need to add support for earlyExitException debug hook !>
   }
   cnt<decisionNumber>++;
} while (true);
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
loop<decisionNumber>:
do {
   var alt<decisionNumber>:int=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:{a | <altSwitchCase(i, a)>}>
 default :
     break loop<decisionNumber>;
   }
} while (true);
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|)
 by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
```

```
altSwitchCase(altNum, alt) ::= <<
case <altNum> :
   <@prealt()>
   <alt>
   break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions)
 ::= <<
<if(label)><label>=<labelType>(<endif>matchStream(input,<token>,FOLLOW_<token>_in_<ruleName><element
Index>)<if(label)>)<endif>; <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(token,label,elementIndex,terminalOptions)>
<listLabel(label, label)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new Array();
list_<label>.push(<elem>);<\n>
>>
```

```
/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label>=<labelType>(input.LT(1));<\n>
<endif>
<endif>
if
 ( <s> ) {
   input.consume();
   <postmatchCode>
<if(!LEXER)>
   this.state.errorRecovery=false;
<endif>
   <if(backtracking)>this.state.failed=false;<endif>
}
else {
   <ruleBacktrackFailure()>
   <@mismatchedSetException()>
<if(LEXER)>
   throw recover(new MismatchedSetException(null,input));<\n>
<else>
   throw new MismatchedSetException(null,input);
   <! use following code to make it recover inline; remove throw mse;
   recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
   !>
<endif>
}<\n>
>>
```

```
matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(label, label)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex="0") ::= <<
<if(label)>
var <label>Start:int = charIndex;
matchString(<string>); <checkRuleBacktrackFailure()>
var <label>StartLine<elementIndex>:int = line;
var <label>StartCharPos<elementIndex>:int =
 charPositionInLine;
<label> = CommonToken.createFromStream(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start, charIndex-1);
<label>.line = <label>StartLine<elementIndex>;
<label>.charPositionInLine = <label>StartCharPos<elementIndex>;
<else>
matchString(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=<labelType>(input.LT(1));<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(label, label)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(label, elementIndex)>
<listLabel(label, label)>
```

```
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value
 or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*
* GMS: Note:  do not use post-decrement operator!  ASC produces bad code for exceptions in this case.
*      See: https://bugs.adobe.com/jira/browse/ASC-3625
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
state._fsp = state._fsp - 1;
<checkRuleBacktrackFailure()>
>>



/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, label)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope)
 ::= <<
<if(label)>
var <label>Start<elementIndex>:int = charIndex;
var <label>StartLine<elementIndex>:int = line;
var <label>StartCharPos<elementIndex>:int = charPositionInLine;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = CommonToken.createFromStream(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, charIndex-1);
<label>.line = <label>StartLine<elementIndex>;
<label>.charPositionInLine = <label>StartCharPos<elementIndex>;
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
```

```
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(rule,label,args,elementIndex,scope)>
<listLabel(label, label)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
var <label>Start<elementIndex>:int
 = charIndex;
var <label>StartLine<elementIndex>:int = line;
var <label>StartCharPos<elementIndex>:int = charPositionInLine;
match(EOF); <checkRuleBacktrackFailure()>
var <label>:<labelType> = CommonToken.createFromStream(input, EOF,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, charIndex-1);
<label>.line = <label>StartLine<elementIndex>;
<label>.charPositionInLine = <label>StartCharPos<elementIndex>;
<else>
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

// used for left-recursive rules
recRuleDefArg()                       ::= "var <recRuleArg()>:int"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::=
 <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
   matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
```

```
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(pred,description)>) ) {
   <ruleBacktrackFailure()>
   throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges;
 separator="\nelse ">
else {
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   throw new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
   <! Need to add hook for noViableAltException() !>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so
 that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
```

```
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
  alt<decisionNumber>=<eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
  <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.
  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
  alt<decisionNumber>=<eotPredictsAlt>;
<else>
  <ruleBacktrackFailure()>
  throw new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
  <! Need to add hook for noViableAltException !>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
   alt<decisionNumber>=<eotPredictsAlt>;
   break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it |case <it>:}; separator="\n">
    {
   <targetState>
   }
   break;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input);
>>

cyclicDFACtor(dfa) ::= <<

dfa<dfa.decisionNumber> = new DFA(this, <dfa.decisionNumber>,
           "<dfa.description>",
           DFA<dfa.decisionNumber>_eot, DFA<dfa.decisionNumber>_eof, DFA<dfa.decisionNumber>_min,
           DFA<dfa.decisionNumber>_max, DFA<dfa.decisionNumber>_accept, DFA<dfa.decisionNumber>_special,
           DFA<dfa.decisionNumber>_transition<if(dfa.specialStateSTs)>,
DFA<dfa.decisionNumber>_specialStateTransition<endif>);

>>
/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple,
 concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<

private const DFA<dfa.decisionNumber>_eot:Array =
  DFA.unpackEncodedString("<dfa.javaCompressedEOT; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_eof:Array =
  DFA.unpackEncodedString("<dfa.javaCompressedEOF; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_min:Array =
  DFA.unpackEncodedString("<dfa.javaCompressedMin; wrap="\"+\n    \"">", true);
private const DFA<dfa.decisionNumber>_max:Array =
  DFA.unpackEncodedString("<dfa.javaCompressedMax; wrap="\"+\n    \"">", true);
private const DFA<dfa.decisionNumber>_accept:Array =
  DFA.unpackEncodedString("<dfa.javaCompressedAccept; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_special:Array =
  DFA.unpackEncodedString("<dfa.javaCompressedSpecial;
 wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_transition:Array = [
    <dfa.javaCompressedTransition:{s|DFA.unpackEncodedString("<s; wrap="\"+\n\"">")}; separator=",\n">
];
<if(dfa.specialStateSTs)>
  private function DFA<dfa.decisionNumber>_specialStateTransition(dfa:DFA, s:int, _input:IntStream):int {
    <if(LEXER)>
    var input:IntStream = _input;
    <endif>
    <if(PARSER)>
    var input:TokenStream = TokenStream(_input);
    <endif>
    <if(TREE_PARSER)>
    var input:TreeNodeStream = TreeNodeStream(_input);
    <endif>
   var _s:int = s;
    switch ( s ) {
    <dfa.specialStateSTs:{state |
    case <i0> : <! compressed special state numbers 0..n-1 !>
        <state>}; separator="\n">
    }
<if(backtracking)>
    if (this.state.backtracking>0) {this.state.failed=true; return -1;}<\n>
<endif>
    throw dfa.error(new NoViableAltException(dfa.description, <dfa.decisionNumber>,
 _s, input));
  }<\n>
<endif>

protected var dfa<dfa.decisionNumber>:DFA;  // initialized in constructor
```

```
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
var index<decisionNumber>_<stateNumber>:int = input.index;
input.rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/**
 An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(pred,{})>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"
```

```
lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atomAsInt>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atomAsInt>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<stateNumber> >= <lowerAsInt> && LA<decisionNumber>_<stateNumber> \<=
<upperAsInt>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt)
 ::= "(input.LA(<k>) >= <lowerAsInt> && input.LA(<k>) \<= <upperAsInt>)"

setTest(ranges) ::= <<
<ranges; separator="||">
>>

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected var <scope.name>_stack:Array = new Array();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected var <scope.name>_stack:Array = new Array();<\n>
<endif>
>>

returnStructName() ::= "<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>
```

```
/** Generate the Java type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<returnStructName()>
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName(d)
 ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<asTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
var <label.label.text>:<ruleLabelType(referencedRule=label.referencedRule)> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public static class <returnType()> extends <if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope {
   <scope.attributes:{it |public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
};
<endif>
>>

parameterScope(scope)
```

```
 ::= <<
<scope.attributes:{it |<it.name>:<it.type>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
<scope>_stack[<scope>_stack.length-<negIndex>-1].<attr.name>
<else>
<if(index)>
<scope>_stack[<index>].<attr.name>
<else>
<scope>_stack[<scope>_stack.length-1].<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
<scope>_stack[<scope>_stack.length-<negIndex>-1].<attr.name> =<expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name> =<expr>;
<else>
<scope>_stack[<scope>_stack.length-1].<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope)
 ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?<scope>.values.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
```

```
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.values.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.values.<attr.name> =<expr>;
<else>
<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.text:null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.type:0)"
tokenLabelPropertyRef_line(scope,attr)
 ::= "(<scope>!=null?<scope>.line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.charPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.tokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int(<scope>.text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=null?<labelType>(<scope>.start):null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=null?<labelType>(<scope>.stop):null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>!=null?<ASTLabelType>(<scope>.tree):null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(<scope>!=null?(input.tokenStream.toStringWithRange(
 input.treeAdaptor.getTokenStartIndex(<scope>.start),
 input.treeAdaptor.getTokenStopIndex(<scope>.start))):null)
<else>
(<scope>!=null?input.toStringWithTokenRange(<scope>.start,<scope>.stop):null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr)
```

```
 ::= "(<scope>!=null?<scope>.st:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(<scope>!=null?<scope>.type:0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(<scope>!=null?<scope>.lien:0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
   "(<scope>!=null?<scope>.charPositionInLine:0)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(<scope>!=null?<scope>.channel:0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::=
   "(<scope>!=null?<scope>.tokenIndex:0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
   "(<scope>!=null?<scope>.text:null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
   "(<scope>!=null?int(<scope>.text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "<labelType>(retval.start)"
rulePropertyRef_stop(scope,attr) ::= "<labelType>(retval.stop)"
rulePropertyRef_tree(scope,attr)
 ::= "<ASTLabelType>(retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.tokenStream.toStringWithRange(
 input.treeAdaptor.getTokenStartIndex(retval.start),
 input.treeAdaptor.getTokenStopIndex(retval.start))
<else>
input.toStringWithTokenRange(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(charIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st
```

```
 and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static const <name>:BitSet = new BitSet([<words64:{it |<it>};separator=", ">]);<\n>
>>

codeFileExtension() ::= ".as"

true_value() ::= "true"
false_value() ::= "false"
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ActionScript/ActionScript.stg

No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/

finishedBacktracking(block) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
    <block>
<else>
<block>
<endif>
<endif>
>>

@ruleBody.setErrorReturnValue() ::= <<
retval.tree = self._adaptor.errorNode(self.input, retval.start, self.input.LT(-1), re)
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
```

```
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/**
 label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate
```

```
 that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <%
<super.matchSet(postmatchCode={<finishedBacktracking({self._adaptor.addChild(root_0,
<createNodeFromToken(...)>)})>}, ...)>
%>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
<super.matchSet(postmatchCode={<finishedBacktracking({root_0 =
self._adaptor.becomeRoot(<createNodeFromToken(...)>, root_0)})>}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({self._adaptor.addChild(root_0,
 <label>.tree)})>
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({root_0 = self._adaptor.becomeRoot(<label>.tree, root_0)})>
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>
```

```
/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
self._adaptor.addChild(root_0,
 <label>_tree)
})>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

createNodeFromToken(label,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
<terminalOptions.node>(<label>) <! new MethodNode(IDLabel) !>
<else>
self._adaptor.createWithPayload(<label>)
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
self._adaptor.setTokenBoundaries(retval.tree, retval.start, retval.stop)
})>
```

>>

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/ASTParser.stg
No license file was found, but licenses were detected in source scan.

```
@outputFile.imports() ::= <<
<@super.imports()>

<if(!TREE_PARSER)>
<! tree parser would already have imported !>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
>>
```

```
@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

parserCtorBody() ::=
 <%
<super.parserCtorBody()><\n>
TreeAdaptor =
<if(actions.(actionScope).treeAdaptorInitializer)>
 <actions.(actionScope).treeAdaptorInitializer>
<else>
 new <actions.(actionScope).treeAdaptorType; null="CommonTreeAdaptor">()
<end>
;
%>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
private <treeAdaptorType()> adaptor;

public <treeAdaptorType()> TreeAdaptor
{
 get
 {
  return adaptor;
 }

 set
 {
  this.adaptor = value;
  <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
}
>>

treeAdaptorType() ::= <<
<actions.(actionScope).treeAdaptorType; null="ITreeAdaptor">
>>

ruleReturnBaseType() ::= <%
Ast<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<ASTLabelType>, <labelType>>
%>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
```

```
<ASTLabelType> root_0 = default(<ASTLabelType>);<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
 :{it|<ASTLabelType>
 <it.label.text>_tree = default(<ASTLabelType>);}; separator="\n">
<ruleDescriptor.tokenListLabels:{it|<ASTLabelType> <it.label.text>_tree = default(<ASTLabelType>);};
separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
 :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
 :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)adaptor.Nil();
<endif>
<endif>
<endif>
>>

// T
 racking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
```

```
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<token>.Add(<label>);
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/**
 rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule.name>.Add(<label>.Tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>stream_<rule>.Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep,
 // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::= <<
<\n>{
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if (<actions.(actionScope).synpredgate>) {
<endif>
<prevRuleRootRef()>.Tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.Nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER&&rewriteMode)>
<prevRuleRootRef()>.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
if
 (<prevRuleRootRef()>.Tree != null)
 input.ReplaceChildren(adaptor.GetParent(retval.Start), adaptor.GetChildIndex(retval.Start),
adaptor.GetChildIndex(_last), retval.Tree);
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER||!rewriteMode)>
<prevRuleRootRef()>.Tree = root_0;
<endif>
<if(backtracking)>
}
<endif>
}

>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{it|RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
   separator="\n"
```

```
>
<referencedWildcardLabels
 :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
 separator="\n"
>
<referencedWildcardListLabels
 :{it|RewriteRuleSubtreeStream stream_<it>=new
 RewriteRuleSubtreeStream(adaptor,"wildcard <it>",list_<it>);};
 separator="\n"
>
<referencedRuleLabels
   :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.Tree:null);};
   separator="\n"
>
<referencedRuleListLabels
   :{it|RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it>",list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if (<referencedElementsDeep:{el | stream_<el>.HasNext}; separator="||">)
{
 <alt>
}
<referencedElementsDeep:{el | stream_<el>.Reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,
 // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if (!(<referencedElements:{el | stream_<el>.HasNext}; separator="||">))
{
 throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
 <a.alt>
}
<else>
{
 <a.alt>
}
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType>
 root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
```

```
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <%
<@pregen()>
DebugLocation(<e.line>, <e.pos>);<\n>
<e.el>
%>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,args,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(),
 root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,args,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,elementIndex,terminalOptions={}) ::= <<
adaptor.AddChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,elementIndex,terminalOptions={}) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>
```

```
/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting
 in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<rule>.NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<rule>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.AddChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label)
 ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
```

```
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

createImaginaryNode(tokenType,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
<! new MethodNode(IDLabel, args) !>
new <terminalOptions.node>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
%>

createRewriteNodeFromElement(token,args,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new <terminalOptions.node>(stream_<token>.NextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create
 new node from old !>
adaptor.Create(<token>, <args; separator=", ">)
<else>
stream_<token>.NextNode()
<endif>
<endif>
%>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/AST.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products

```
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports()>
import stringtemplate3
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnInit() ::= <<
self.st = None
>>

@returnScope.ruleReturnMembers() ::= <<
def getTemplate(self):
   return self.st

def toString(self):
   if self.st is not None:
       return self.st.toString()
   return None
__str__
 = toString

>>

@genericParser.init() ::= <<
<@super.init()>
self.templateLib = stringtemplate3.StringTemplateGroup(
   '<name>Templates', lexer='angle-bracket'
   )

>>
```

```
@genericParser.members() ::= <<
<@super.members()>
def setTemplateLib(self, templateLib):
   self.templateLib = templateLib

def getTemplateLib(self):
   return self.templateLib

>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.st})>
>>

rewriteTemplate(alts) ::= <<
# TEMPLATE REWRITE
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
   <first(alts):rewriteTemplateAltFirst()>
   <rest(alts):{it | el<rewriteTemplateAlt(it)>}>
   <if(rewriteMode)><replaceTextInLine()><endif>

<else>
<first(alts):rewriteTemplateAltFirst()>
<rest(alts):{it | el<rewriteTemplateAlt(it)>}>
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
self.input.getTokenStream().replace(

  self.input.getTreeAdaptor().getTokenStartIndex(retval.start),
   self.input.getTreeAdaptor().getTokenStopIndex(retval.start),
   retval.st
   )
<else>
self.input.replace(
   retval.start.getTokenIndex(),
   self.input.LT(-1).getTokenIndex(),
   retval.st
   )
<endif>
>>
```

```
rewriteTemplateAltFirst(alt) ::= <<
<if(alt.pred)>
if <alt.pred>:
   # <alt.description>
   retval.st = <alt.alt>
<\n>
<else>
# <alt.description>
retval.st = <alt.alt>
<\n>
<endif>
>>

rewriteTemplateAlt(alt) ::= <<
<if(alt.pred)>if <alt.pred>:
   # <alt.description>
   retval.st = <alt.alt>
<\n>
<else>se:
   # <alt.description>
   retval.st = <alt.alt>
<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
None
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args)
 ::= <%
self.templateLib.getInstanceOf("<name>"<if(args)>, attributes={<args:{a | "<a.name>": <a.value>}; separator=",
">}<endif>)
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
self.templateLib.getInstanceOf(<expr><if(args)>, attributes={<args:{a | "<a.name>": <a.value>}; separator=",
">}<endif>)
%>
```

```
/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <%
stringtemplate3.StringTemplate("<template>", group=self.templateLib<if(args)>, attributes={<args:{a |
"<a.name>": <a.value>}; separator=", ">}<endif>)
%>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>)["<attrName>"] = <expr>
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
stringtemplate3.StringTemplate(<stringExpr>, group=self.templateLib)
>>
```

Found
in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/ST.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2012 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/

finishedBacktracking(block) ::= <<
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
   <block>
<else>
<block>
<endif>
>>

/**
 Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_first_0 = None
_last = None<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<finishedBacktracking({
<if(rewriteMode)>
retval.tree = _first_0
if self._adaptor.getParent(retval.tree) is not None and self._adaptor.isNil(self._adaptor.getParent(retval.tree)):
   retval.tree = self._adaptor.getParent(retval.tree)
<endif>
})>
```

```
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = self.input.LT(1)
_save_last_<treeLevel> = _last
_first_<treeLevel> = None
<if(!rewriteMode)>
root_<treeLevel> = self._adaptor.nil()<\n>
<endif>
<root:element()>
<if(rewriteMode)>
<finishedBacktracking({
<if(root.el.rule)>
if _first_<enclosingTreeLevel>
 is None:
   _first_<enclosingTreeLevel> = <root.el.label>.tree<\n>
<else>
if _first_<enclosingTreeLevel> is None:
   _first_<enclosingTreeLevel> = <root.el.label><\n>
<endif>
})>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if self.input.LA(1) == DOWN:
   self.match(self.input, DOWN, None)
   <children:element()>
   self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)<\n>
<endif>
<if(!rewriteMode)>
self._adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>)<\n>
<endif>
_last = _save_last_<treeLevel>

>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
```

```
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree
 = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label><\n>
})>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
```

```
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={})
 ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.wildcard(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<label>_tree = self._adaptor.dupTree(<label>)
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label>
})>
<endif>
>>

// SET AST
matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<endif>
}, ...)>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={})
 ::= <<
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
```

```
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
}, ...)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRef(...)>
<finishedBacktracking({
<if(!rewriteMode)>
self._adaptor.addChild(root_<treeLevel>, <label>.tree)
<else> <! rewrite mode !>
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label>.tree<\n>
<endif>
})>
>>

/** x+=rule auto construct
 */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
```

```
<super.ruleRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
root_<treeLevel> = self._adaptor.becomeRoot(<label>.tree, root_<treeLevel>)
})>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope)
 ::= <<
_last = self.input.LT(1)
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,scope,terminalOptions={}) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
```

```
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
})>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python3/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/

parserMembers() ::= <<
 // Implement this function in your helper file to use a custom tree adaptor
 partial void InitializeTreeAdaptor();
 protected DebugTreeAdaptor adaptor;

 public
 ITreeAdaptor TreeAdaptor
 {
  get
  {
   return adaptor;
  }
  set
  {
<if(grammar.grammarIsRoot)>
   this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
   this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
   <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
  }
 }<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port,
<if(TREE_PARSER)>input.TreeAdaptor<else>adaptor<endif> );
DebugListener = proxy;
<inputStreamType> = new Debug<inputStreamType>( input, proxy );
try
{
 proxy.Handshake();
}
catch ( IOException ioe )
{
 ReportError( ioe );
}
>>
```

```
@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

@ctorForProfilingRootGrammar.finally() ::=<<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
>>

@ctorForPredefinedListener.superClassRef()
 ::= ": base( input, dbg )"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)><! don't create new adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>

//@rewriteElement.pregen() ::= "dbg.Location( <e.line>, <e.pos> );"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/
@rule.setErrorReturnValue() ::= <<
/* ASTParser rule.setErrorReturnValue */
retval.tree
 = (<ASTLabelType> *)[treeAdaptor errorNode:input From:retval.start To:[input LT:-1] Exception:re];
<! System.out.println("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token, label, elementIndex, terminalOptions) ::= <<
/* ASTParser tokenRef */
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
[treeAdaptor addChild:<label>_tree  toTree:root_0];
<if(backtracking)>}<endif>
>>

/* ID! and output=AST (same as plain tokenRef) */
/* ASTParser tokenRefBang */
tokenRefBang(token,label,elementIndex,terminalOptions) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>_tree old:root_0];
<if(backtracking)>}<endif>
>>

/**
 ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser tokenRefBangAndListLabel */
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser tokenRefAndListLabel */
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,terminalOptions,elementIndex) ::= <<
/* ASTParser tokenRefRuleRootAndListLabel */
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template
 name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,terminalOptions,elementIndex,postmatchCode) ::= <%
/* ASTParser matchSet */
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
   [treeAdaptor addChild:<createNodeFromToken(...)> toTree:root_0 ];}, ...)>
%>

matchRuleBlockSet(s,label,terminalOptions,elementIndex,postmatchCode,treeLevel="0") ::= <<
```

```
/* ASTParser matchRuleBlockSet */
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,terminalOptions, postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,terminalOptions,elementIndex,debug)
 ::= <<
/* ASTParser matchSetRuleRoot */
<if(label)>
<label>=(<labelType> *)[input LT:1]; /* matchSetRuleRoot */<\n>
<endif>
<super.matchSet(postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:<createNodeFromToken(...)> old:root_0];}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRef */
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
[treeAdaptor addChild:[<label> getTree] toTree:root_0];
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefRuleRoot */
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:[<label>
 getTree] old:root_0];
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefAndListLabel */
<ruleRef(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>
```

```
/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefBangAndListLabel */
<ruleRefBang(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
/* ASTParser ruleRefRuleRootAndListLabel */
<ruleRefRuleRoot(...)>
<listLabel(elem = {[<label> getTree]},...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser wildcard */
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
  [treeAdaptor addChild:[[treeAdaptor create:<label>] retain] toTree:root_0];
<if(backtracking)>}<endif>
>>

wildcardBang(token,label,elementIndex,terminalOptions)
 ::= "<super.wildcard(...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions) ::= <<
/* ASTParser wildcardRuleRoot */
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
  <label>_tree = [[treeAdaptor create:<label>] retain]
  root_0 = (<ASTLabelType> *)[treeAdaptor becomeRoot:<label>_tree old:root_0];
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,terminalOptions) ::= <<
/* ASTParser createNodeFromToken */
<if(terminalOptions.node)>
[<terminalOptions.node> new<terminalOptions.node>:<label>] <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType> *)[[treeAdaptor create:<label>] retain]
<endif>
>>

// straight from java cleanup ///
ruleCleanUp() ::= <<
/* ASTParser ruleCleanUp */
```

```
<super.ruleCleanUp()>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
   retval.tree = (<ASTLabelType> *)[treeAdaptor rulePostProcessing:root_0];
   [treeAdaptor setTokenBoundaries:retval.tree
 From:retval.start To:retval.stopToken];
<if(backtracking)>}<endif>
>>
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/ObjC/ASTParser.stg
No license file was found, but licenses were detected in source scan.

```
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/

@outputFile.imports() ::= <<
<@super.imports()>
import org.antlr.stringtemplate.*;
import org.antlr.stringtemplate.language.*;
import java.util.HashMap;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
public StringTemplate st;
public Object getTemplate() { return st; }
public String toString() { return st==null?null:st.toString();
 }
>>

@genericParser.members() ::= <<
<@super.members()>
 protected StringTemplateGroup templateLib =
   new StringTemplateGroup("<name>Templates", AngleBracketTemplateLexer.class);

 public void setTemplateLib(StringTemplateGroup templateLib) {
   this.templateLib = templateLib;
 }
 public StringTemplateGroup getTemplateLib() {
   return templateLib;
 }
 /** allows convenient multi-value initialization:
  *  "new STAttrMap().put(...).put(...)"
  */
 @SuppressWarnings("serial")
 public static class STAttrMap extends HashMap\<String, Object> {
  public STAttrMap put(String attrName, Object value) {
   super.put(attrName, value);
   return this;
  }
 }
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(rule,label,elementIndex,args,scope)>
<listLabel(label, {<label>.getTemplate()})>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
 <alts:rewriteTemplateAlt();
 separator="else ">
 <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
((TokenRewriteStream)input.getTokenStream()).replace(
 input.getTreeAdaptor().getTokenStartIndex(retval.start),
 input.getTreeAdaptor().getTokenStopIndex(retval.start),
 retval.st);
<else>
((TokenRewriteStream)input).replace(
 ((Token)retval.start).getTokenIndex(),
 input.LT(-1).getTokenIndex(),
 retval.st);
<endif>
>>

rewriteTemplateAlt(alt) ::= <<
// <alt.description>
<if(alt.pred)>
if (<alt.pred>) {
 retval.st = <alt.alt>;
}<\n>
<else>
{
 retval.st = <alt.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
null;
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template;
 you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args) ::= <%
```

```
templateLib.getInstanceOf("<name>"<if(args)>,
 new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
 <endif>)
%>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <%
templateLib.getInstanceOf(<expr><if(args)>,
 new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
 <endif>)
%>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <%
new StringTemplate(templateLib, "<template>"<if(args)>,
 new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
 <endif>)
%>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>).setAttribute("<attrName>",<expr>);
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(templateLib,<stringExpr>)
>>
```

Found
 in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Java/ST.stg
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2011 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
```

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid.
 There is no huge explosion of combinations.
*/

@rule.setErrorReturnValue() ::= <<
retval.Tree = (<ASTLabelType>)adaptor.ErrorNode(input, retval.Start, input.LT(-1), re);
<! System.out.WriteLine("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <%
<super.tokenRef(...)>
```

```
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)><\n>if (state.backtracking == 0) {<endif>
<\n><label>_tree = <createNodeFromToken(...)>;
<\n>adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)><\n>}<endif>
<endif>
%>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <%
<super.tokenRef(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)><\n>if (<actions.(actionScope).synpredgate>) {<endif>
<\n><label>_tree =
 <createNodeFromToken(...)>;
<\n>root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)><\n>}<endif>
<endif>
%>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending
```

```
 on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
<super.matchSet(postmatchCode={<if(!ruleDescriptor.isSynPred)><if(backtracking)>if
(<actions.(actionScope).synpredgate>) <endif>adaptor.AddChild(root_0, <createNodeFromToken(...)>);<endif>},
...)>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts:
 (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);
<endif>
<super.matchSet(postmatchCode={<if(!ruleDescriptor.isSynPred)><if(backtracking)>if
(<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<createNodeFromToken(...)>, root_0);<endif>}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <%
<super.ruleRef(...)>
<if(!ruleDescriptor.isSynPred)>
<\n><if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>adaptor.AddChild(root_0, <label>.Tree);
<endif>
%>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
```

```
<super.ruleRef(...)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_0);
>>

/** x+=rule when
 output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabelElem(elem={<label>.Tree},elemType=ASTLabelType,...)>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(token=[],...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={})
 ::= <<
<super.wildcard(...)>
<if(!ruleDescriptor.isSynPred)>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
<endif>
>>
```

```
createNodeFromToken(label,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
new
<terminalOptions.node>(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalOptions.tex
t)>,<terminalOptions.text; format="string"><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<if(terminalOptions.type)><terminalOptions.type>,<endif><label><if(terminalO
ptions.text)>,<terminalOptions.text; format="string"><endif>)
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>if (<actions.(actionScope).synpredgate>) {<endif>
retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
adaptor.SetTokenBoundaries(retval.Tree, retval.Start, retval.Stop);
<if(backtracking)>}<endif>
>>
```

Found
in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/ASTParser.stg
No license file was found, but licenses were detected in source scan.

```
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/ToolSTGroupFile.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2011 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2011 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED
 WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

csharpVisibilityMap ::= [
 "private":"private",
 "protected":"protected",
 "public":"public",
 "fragment":"private",
 default:"private"
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile( LEXER,PARSER,TREE_PARSER, actionScope, actions,
   docComment, recognizer,
   name,
 tokens, tokenNames, rules, cyclicDFAs,
   bitsets, buildTemplate, buildAST, rewriteMode, profile,
   backtracking, synpreds, memoize, numRules,
   fileName, ANTLRVersion, generatedTimestamp, trace,
   scopes, superClass, literals) ::=
<<
//------------------------------------------------------------------------------
// \<auto-generated>
//     This code was generated by a tool.
//     ANTLR Version: <ANTLRVersion>
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// \</auto-generated>
//------------------------------------------------------------------------------

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

// The variable 'variable' is assigned but its value is never used.
#pragma warning disable 219
// Unreachable code detected.
#pragma warning disable 162
// Missing XML comment for publicly visible type or member 'Type_or_Member'
#pragma warning disable 1591
// CLS compliance checking will not
 be performed on 'type' because it is not visible from outside this assembly.
#pragma warning disable 3019

<actions.(actionScope).header>

<@imports>
using System.Collections.Generic;
using Antlr.Runtime;
using Antlr.Runtime.Misc;
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
```

```
<endif>
<@end>
<if(actions.(actionScope).namespace)>
namespace <actions.(actionScope).namespace>
{
<endif>
<docComment>
<recognizer>
<if(actions.(actionScope).namespace)>

} // namespace <actions.(actionScope).namespace>
<endif>
>>

lexerInputStreamType() ::= <<
<actions.(actionScope).inputStreamType; null="ICharStream">
>>

lexer(grammar, name, tokens, scopes, rules, numRules, filterMode, labelType="CommonToken",
superClass={<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Lexer
<endif>},
   rewriteElementType={}, ASTLabelType={}) ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar,
 actions=actions)> partial class <grammar.recognizerName> : <@superClassName><superClass><@end>
{
 <tokens:{it|public const int <it.name; format="id">=<it.type>;}; separator="\n">
 <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <actions.lexer.members>

 // delegates
 <grammar.delegates:
   {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
 // delegators
 <grammar.delegators:
   {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
 <last(grammar.delegators):{g|private <g.recognizerName> gParent;}>

 <actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>()<! needed by subclasses !>
 {
  OnCreated();
 }

 <actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType()>
input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}> )
  : this(input, new RecognizerSharedState()<grammar.delegators:{g|,
 <g:delegateName()>}>)
 {
 }

 <actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<lexerInputStreamType()> input,
RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
  : base(input, state)
 {
<if(memoize)>
<if(grammar.grammarIsRoot)>
  state.ruleMemo = new System.Collections.Generic.Dictionary\<int, int>[<numRules>+1];<\n><! index from 1..n !>
<endif>
<endif>
  <grammar.directDelegates:
   {g|<g:delegateName()> = new <g.recognizerName>(input, this.state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
  <grammar.delegators:
   {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>

  OnCreated();
 }
 public override string GrammarFileName { get { return "<fileName>"; } }

<if(grammar.hasDelegates)>
 public override <lexerInputStreamType()> CharStream
 {
  get
  {
   return base.CharStream;
  }
  set
  {
   base.CharStream
 = value;
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
   <grammar.delegators:
    {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
   <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
  }
 }

<if(grammar.delegates)>
 public override void SetState(RecognizerSharedState state)
 {
  base.SetState(state);
  <grammar.delegates:{g|<g:delegateName()>.SetState(state);}; separator="\n">
 }
<endif>

<endif>
<if(filterMode)>
 <filteringNextToken()>
<endif>


 partial void OnCreated();
 partial void EnterRule(string ruleName, int ruleIndex);
 partial void LeaveRule(string ruleName, int ruleIndex);

 <rules; separator="\n">

 <insertLexerSynpreds(synpreds)>

 #region DFA
 <cyclicDFAs:{dfa | DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

 protected override void InitDFAs()
 {
  base.InitDFAs();
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber>
 = new DFA<dfa.decisionNumber>(this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif>);}; separator="\n">
 }

 <cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>
 #endregion

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public override IToken NextToken()
{
 while (true)
 {
  if (input.LA(1) == CharStreamConstants.EndOfFile)
  {
   IToken eof = new CommonToken((ICharStream)input, CharStreamConstants.EndOfFile, TokenChannels.Default,
input.Index, input.Index);
   eof.Line = Line;
   eof.CharPositionInLine = CharPositionInLine;
   return eof;
  }
  state.token = null;
  state.channel = TokenChannels.Default;
  state.tokenStartCharIndex
 = input.Index;
  state.tokenStartCharPositionInLine = input.CharPositionInLine;
  state.tokenStartLine = input.Line;
  state.text = null;
  try
  {
   int m = input.Mark();
   state.backtracking=1;<! means we won't throw slow exception !>
   state.failed=false;
   mTokens();
   state.backtracking=0;
   <! mTokens backtracks with synpred at backtracking==2
      and we set the synpredgate to allow actions at level 1. !>
   if (state.failed)
   {
    input.Rewind(m);
    input.Consume();<! advance one char and try again !>
   }
   else
   {
    Emit();
    return state.token;
   }
  }
  catch (RecognitionException re)
  {
   // shouldn't happen in backtracking mode, but...
   ReportError(re);
   Recover(re);
  }
 }
}

public override void Memoize(IIntStream input, int ruleIndex, int ruleStartIndex)
{
 if (state.backtracking > 1)
  base.Memoize(input, ruleIndex, ruleStartIndex);
}
```

```
public override bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
 if (state.backtracking > 1)
  return
 base.AlreadyParsedRule(input, ruleIndex);

 return false;
}
>>

actionGate() ::= "state.backtracking == 0"

filteringActionGate() ::= "state.backtracking == 1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
            labelType, members, rewriteElementType,
            filterMode, ASTLabelType="object") ::= <<
[System.CodeDom.Compiler.GeneratedCode("ANTLR", "<ANTLRVersion>")]
[System.CLSCompliant(false)]
<parserModifier(grammar=grammar, actions=actions)> partial class <grammar.recognizerName> :
<@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
 internal static readonly string[] tokenNames = new string[] {
  "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
 };
<endif>
 <tokens:{it|public const int <it.name; format="id">=<it.type>;}; separator="\n">

<if(grammar.delegates)>
 // delegates
 <grammar.delegates:
   {g|private <g.recognizerName>
 <g:delegateName()>;}; separator="\n">
<endif>
<if(grammar.delegators)>
 // delegators
 <grammar.delegators:
   {g|private <g.recognizerName> <g:delegateName()>;}; separator="\n">
 <last(grammar.delegators):{g|private <g.recognizerName> gParent;}>
<endif>

<if(grammar.delegates)>
 public override void SetState(RecognizerSharedState state)
 {
  base.SetState(state);
  <grammar.delegates:{g|<g:delegateName()>.SetState(state);}; separator="\n">
 }

<if(TREE_PARSER)>
 public override void SetTreeNodeStream(ITreeNodeStream input)
 {
  base.SetTreeNodeStream(input);
  <grammar.delegates:{g|<g:delegateName()>.SetTreeNodeStream(input);}; separator="\n">
 }
<endif>
<endif>

 <scopes:{it|<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <@members()>

 public override string[] TokenNames { get { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; } }
 public override string GrammarFileName { get { return "<fileName>"; } }

 <members>

 partial void
 OnCreated();
 partial void EnterRule(string ruleName, int ruleIndex);
 partial void LeaveRule(string ruleName, int ruleIndex);

 #region Rules
 <rules; separator="\n">
 #endregion Rules

<if(grammar.delegatedRules)>
<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
 #region Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
 <ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <returnType(ruleDescriptor)>
<ruleDescriptor.name; format="id">(<ruleDescriptor.parameterScope:parameterScope()>)<!throws
RecognitionException !>{ <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name;
format="id">(<if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope.attributes:{a|<a.name;
format="id">}; separator=", "><endif>); \}}; separator="\n">
 #endregion Delegated rules
<endif>

 <insertSynpreds(synpreds)>

<if(cyclicDFAs)>
 #region DFA
 <cyclicDFAs:{dfa | private
 DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

 protected override void InitDFAs()
 {
  base.InitDFAs();
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>( this<if(dfa.specialStateSTs)>,
SpecialStateTransition<dfa.decisionNumber><endif> );}; separator="\n">
 }

 <cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>
 #endregion DFA
<endif>

<if(bitsets)>
 #region Follow sets
 private static class Follow
 {
  <bitsets:{it|<bitset(name={_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>}; separator="\n">
 }
 #endregion Follow sets
<endif>
}
>>

@genericParser.members() ::= <<
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
<actions.(actionScope).ctorModifier; null="public"> <grammar.recognizerName>(<inputStreamType>
input<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
 : this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>)
{
}
<actions.(actionScope).ctorModifier; null="public">
 <grammar.recognizerName>(<inputStreamType> input, RecognizerSharedState state<grammar.delegators:{g|,
<g.recognizerName> <g:delegateName()>}>)
 : base(input, state)
{
<if(grammar.directDelegates)>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
<endif>
<if(grammar.indirectDelegates)>
 <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
<endif>
<if(grammar.delegators)>
 <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
```

```
<endif>
 <parserCtorBody()>
 OnCreated();
}
>>

// imported grammars are 'public' (can't be internal because their return scope classes must be accessible)
parserModifier(grammar, actions) ::= <<
<if(grammar.grammarIsRoot)><actions.(actionScope).modifier; null="public"><else>public<endif>
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo
 = new System.Collections.Generic.Dictionary\<int, int>[<length(grammar.allImportedRules)>+1];<\n><! index
from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
      ASTLabelType="object",
superClass={<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Parser
<endif>}, labelType="IToken",
      members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="IToken", filterMode=false, ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
          numRules, bitsets, filterMode, labelType={<ASTLabelType>}, ASTLabelType="object",
superClass={<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else>Antlr.Runtime.Tree.<
if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif><endif>},
           members={<actions.treeparser.members>}) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
```

```
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<

partial void EnterRule_<ruleName>_fragment();
partial void LeaveRule_<ruleName>_fragment();

// $ANTLR start <ruleName>
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> void
<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope()>)
{
 <ruleLabelDefs(...)>
 EnterRule_<ruleName>_fragment();
 EnterRule("<ruleName>_fragment",
 <ruleDescriptor.index>);
 TraceIn("<ruleName>_fragment", <ruleDescriptor.index>);
 try
 {
  <block>
 }
 finally
 {
  TraceOut("<ruleName>_fragment", <ruleDescriptor.index>);
  LeaveRule("<ruleName>_fragment", <ruleDescriptor.index>);
  LeaveRule_<ruleName>_fragment();
 }
}
// $ANTLR end <ruleName>
>>

insertLexerSynpreds(synpreds) ::= <<
<insertSynpreds(synpreds)>
>>

insertSynpreds(synpreds) ::= <<
<if(synpreds)>
#region Synpreds
private bool EvaluatePredicate(System.Action fragment)
{
 bool success = false;
 state.backtracking++;
 <@start()>
 try { DebugBeginBacktrack(state.backtracking);
 int start = input.Mark();
 try
 {
  fragment();
 }
 catch ( RecognitionException re )
 {
  System.Console.Error.WriteLine("impossible: "+re);
 }
 success = !state.failed;
 input.Rewind(start);
 } finally { DebugEndBacktrack(state.backtracking, success); }
 <@stop()>
 state.backtracking--;
 state.failed=false;
 return
 success;
}
#endregion Synpreds
<endif>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if (state.backtracking > 0 && AlreadyParsedRule(input, <ruleDescriptor.index>)) { <returnFromRule()> }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) <returnFromRule()><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true; <returnFromRule()>}<endif>
>>

ruleWrapperMap ::= [
 "bottomup":{<ruleWrapperBottomup()>},
 "topdown":{<ruleWrapperTopdown()>},
 default:""
]

ruleWrapperBottomup() ::= <<
<if(TREE_PARSER && filterMode)>
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif> Bottomup() { <if(buildAST)>return
<endif>bottomup(); }
<endif>
>>

ruleWrapperTopdown() ::= <<
<if(TREE_PARSER && filterMode)>
```

```
protected override <if(buildAST)>IAstRuleReturnScope<else>void<endif>
 Topdown() { <if(buildAST)>return <endif>topdown(); }
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(ruleDescriptor.returnScope)>
partial void EnterRule_<ruleName>();
partial void LeaveRule_<ruleName>();
<ruleWrapperMap.(ruleName)>
// $ANTLR start "<ruleName>"
// <fileName>:<description>
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> <returnType(ruleDescriptor)> <ruleName;
format="id">(<ruleDescriptor.parameterScope:parameterScope()>)
{
 EnterRule_<ruleName>();
 EnterRule("<ruleName>", <ruleDescriptor.index>);
 TraceIn("<ruleName>", <ruleDescriptor.index>);
 <ruleScopeSetUp()>
 <ruleDeclarations()>
 <ruleLabelDefs(...)>
 <ruleDescriptor.actions.init>
 try { DebugEnterRule(GrammarFileName, "<ruleName>");
 DebugLocation(<ruleDescriptor.tree.line>,
 <ruleDescriptor.EORNode.charPositionInLine>);
 <@preamble()>
 try
 {
  <ruleMemoization(name=ruleName)>
  <block>
  <ruleCleanUp()>
  <(ruleDescriptor.actions.after):execAction()>
 }
<if(exceptions)>
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
 <actions.(actionScope).rulecatch>
<else>
 catch (RecognitionException re)
 {
  ReportError(re);
  Recover(input,re);
  <@setErrorReturnValue()>
 }
<endif>
<endif>
<endif>
 finally
 {
  TraceOut("<ruleName>", <ruleDescriptor.index>);
  LeaveRule("<ruleName>", <ruleDescriptor.index>);
  LeaveRule_<ruleName>();
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
 }
 DebugLocation(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);
 } finally { DebugExitRule(GrammarFileName, "<ruleName>"); }
 <@postamble()>
 <returnFromRule()><\n>
}
// $ANTLR end "<ruleName>"
>>

// imported grammars need to have internal
 rules
ruleModifier(grammar,ruleDescriptor) ::= <<
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>internal<endif>
>>

// imported grammars need to have public return scopes
returnScopeModifier(grammar,ruleDescriptor) ::= <<
<if(grammar.grammarIsRoot)><csharpVisibilityMap.(ruleDescriptor.modifier);
null="private"><else>public<endif>
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
 <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType(ruleDescriptor)> retval = new
<returnType(ruleDescriptor)>(<if(ruleDescriptor.returnScope.attributes)>this<endif>);
retval.Start = (<labelType>)input.LT(1);
```

```
<elseif(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name; format="id"> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.Index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{it|<it>_stack.Push(new
 <it>_scope(this));<it>_scopeInit(<it>_stack.Peek());}; separator="\n">
<ruleDescriptor.ruleScope:{it|<it.name>_stack.Push(new
<it.name>_scope(this));<it.name>_scopeInit(<it.name>_stack.Peek());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{it|<it>_scopeAfter(<it>_stack.Peek());<it>_stack.Pop();}; separator="\n">
<ruleDescriptor.ruleScope:{it|<it.name>_scopeAfter(<it.name>_stack.Peek());<it.name>_stack.Pop();};
separator="\n">
>>

ruleLabelDefs(ruleDescriptor, labelType, ASTLabelType, rewriteElementType) ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wil
dcardTreeListLabels]
 :{it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<ruleDescriptor.tokenListLabels
 :{it|List\<<labelType>\> list_<it.label.text> = null;}; separator="\n"
>
<[ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
 :{it|List\<<ASTLabelType>\> list_<it.label.text>
 = null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
<ruleDescriptor.ruleListLabels:ruleLabelDef(); separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
 :{it|<labelType> <it.label.text> = default(<labelType>);}; separator="\n"
>
<[ruleDescriptor.charListLabels,
 ruleDescriptor.charLabels]
 :{it|int <it.label.text> = 0;}; separator="\n"
```

```
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
 :{it|List\<<labelType>\> list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.charListLabels:{it|List\<int> list_<it.label.text> = null;}; separator="\n"
>
>>

returnFromRule() ::= <%
return
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<! This comment is a hack to make sure the following
  single space appears in the output. !> <ruleDescriptor.singleValueReturnName>
<else>
<!!>
 retval
<endif>
<endif>
<endif>
;
%>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.Stop = (<labelType>)input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (state.backtracking > 0) { Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<

partial void EnterRule_<ruleName>();
partial void LeaveRule_<ruleName>();
```

```
// $ANTLR start "<ruleName>"
[GrammarRule("<ruleName>")]
<ruleModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> void
m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>)
{
 EnterRule_<ruleName>();
 EnterRule("<ruleName>", <ruleDescriptor.index>);
 TraceIn("<ruleName>", <ruleDescriptor.index>);
 <ruleScopeSetUp()>
 <ruleDeclarations()>
 try
 {
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
<else>
  int
 _type = <ruleName>;
  int _channel = DefaultTokenChannel;
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
  <ruleCleanUp()>
  state.type = _type;
  state.channel = _channel;
  <(ruleDescriptor.actions.after):execAction()>
<endif>
 }
 finally
 {
  TraceOut("<ruleName>", <ruleDescriptor.index>);
  LeaveRule("<ruleName>", <ruleDescriptor.index>);
  LeaveRule_<ruleName>();
  <ruleScopeCleanUp()>
  <memoize()>
 }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
```

```
public override void mTokens()
{
 <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
//
 <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterSubRule(<decisionNumber>);
try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
{
<alts:{a|<altSwitchCase(i,a)>}>
}
} finally { DebugExitSubRule(<decisionNumber>); }
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);
<decision>
} finally { DebugExitDecision(<decisionNumber>); }
<@postdecision()>
switch (alt<decisionNumber>)
{
<alts:{a|<altSwitchCase(i,a)>}>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description)
```

```
 ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
DebugEnterAlt(1);
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);
 <decision>
 } finally { DebugExitDecision(<decisionNumber>); }
 <@postdecision()>
 switch (alt<decisionNumber>)
 {
 <alts:{a|<altSwitchCase(i,a)>}>
 default:
  if (cnt<decisionNumber> >= 1)
   goto loop<decisionNumber>;

  <ruleBacktrackFailure()>
  EarlyExitException
 eee<decisionNumber> = new EarlyExitException( <decisionNumber>, input );
  DebugRecognitionException(eee<decisionNumber>);
  <@earlyExitException()>
  throw eee<decisionNumber>;
 }
 cnt<decisionNumber>++;
}
loop<decisionNumber>:
 ;

} finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
try { DebugEnterSubRule(<decisionNumber>);
while (true)
{
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 try { DebugEnterDecision(<decisionNumber>, false<!<decision.dfa.hasSynPred>!>);
 <decision>
 } finally { DebugExitDecision(<decisionNumber>); }
 <@postdecision()>
 switch ( alt<decisionNumber> )
 {
 <alts:{a|<altSwitchCase(i,a)>}>
 default:
  goto loop<decisionNumber>;
 }
}

loop<decisionNumber>:
 ;

}
 finally { DebugExitSubRule(<decisionNumber>); }
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
```

```
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
 <@prealt()>
 DebugEnterAlt(<altNum>);
 <alt>
 break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel,
 treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element(it) ::= <%
<@prematch()>
DebugLocation(<it.line>, <it.pos>);<\n>
<it.el><\n>
%>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<if(label)><label>=(<labelType>)<endif>Match(input,<token>,Follow._<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>
```

```
/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

listLabel(label,elem) ::= <<
#error The listLabel template should not be used with this target.<\n>
>>

listLabelElem(label,elem,elemType) ::= <<
if (list_<label>==null) list_<label>=new List\<<elemType; null={<labelType>}>\>();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>

/**
 match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="",terminalOptions={}) ::= <<
<if(label)>
<matchSetLabel()>
<endif>
if (<s>)
{
 input.Consume();
 <postmatchCode>
 <if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
}
else
{
 <ruleBacktrackFailure()>
 MismatchedSetException mse = new MismatchedSetException(null,input);
 DebugRecognitionException(mse);
 <@mismatchedSetException()>
<if(LEXER)>
 Recover(mse);
 throw mse;
<else>
 throw mse;
 <! use following code to make it recover inline; remove throw mse;
 recoverFromMismatchedSet(input,mse,Follow._set_in_<ruleName><elementIndex>);
 !>
<endif>
}<\n>
>>

matchSetUnchecked(s,label,elementIndex,postmatchCode=false) ::= <%
<if(label)>
<matchSetLabel()><\n>
<endif>
input.Consume();<\n>
<if(postmatchCode)>
<postmatchCode><\n>
<endif>
<if(!LEXER)>state.errorRecovery=false;<endif><if(backtracking)>state.failed=false;<endif>
%>

matchSetLabel()
 ::= <%
<if(LEXER)>
<label>= input.LA(1);
<else>
<label>=(<labelType>)input.LT(1);
<endif>
%>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <%
<if(label)>
int <label>Start = CharIndex;<\n>
Match(<string>); <checkRuleBacktrackFailure()><\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<label> = new <labelType>(input, TokenTypes.Invalid, TokenChannels.Default, <label>Start, CharIndex-1);<\n>
```

```
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
%>

wildcard(token,label,elementIndex,terminalOptions={})
 ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<wildcard(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(Follow._<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name;
 format="id">(<args; separator=", ">);
PopFollow();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
```

```
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabelElem(elem=label,elemType={<ASTLabelType>},...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()><\n>
<label> = new <labelType>(input, TokenTypes.Invalid, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line
 = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
%>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabelElem(elem=label,elemType=labelType,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <%
<if(label)>
int <label>Start<elementIndex> = CharIndex;<\n>
int <label>StartLine<elementIndex> = Line;<\n>
int <label>StartCharPos<elementIndex> = CharPositionInLine;<\n>
Match(EOF); <checkRuleBacktrackFailure()><\n>
<labelType> <label> = new <labelType>(input, EOF, TokenChannels.Default, <label>Start<elementIndex>,
CharIndex-1);<\n>
<label>.Line = <label>StartLine<elementIndex>;<\n>
<label>.CharPositionInLine = <label>StartCharPos<elementIndex>;
<else>
Match(EOF); <checkRuleBacktrackFailure()>
```

```
<endif>
%>

// used for left-recursive
 rules
recRuleDefArg()                       ::= "int <recRuleArg()>"
recRuleArg()                          ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()              ::= "root_0=$<ruleName>_primary.tree;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (input.LA(1) == TokenTypes.Down)
{
 Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenTypes.Down, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenTypes.Up, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description)
 ::= <<
if (!(<evalPredicate(...)>))
{
 <ruleBacktrackFailure()>
 throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
{
```

```
<if(eotPredictsAlt)>
 alt<decisionNumber> = <eotPredictsAlt>;
<else>
 <ruleBacktrackFailure()>
 NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
 DebugRecognitionException(nvae);
 <@noViableAltException()>
 throw nvae;
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<k> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt>;<! if no edges, don't gen ELSE !>
<else>
else
{
 alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>;"
```

```
/** A simple edge with an expression.  If the expression is satisfied,
*  enter
 to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> && (<predicates>)<endif>)
{
 <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
 alt<decisionNumber>=<eotPredictsAlt>;
 break;<\n>
<else>
 {
  <ruleBacktrackFailure()>
  NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input, <k>);
  DebugRecognitionException(nvae);
  <@noViableAltException()>
  throw nvae;
 }
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
{
<edges;
 separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch (input.LA(<k>))
```

```
{
<edges; separator="\n">
<if(eotPredictsAlt)>
default:
 alt<decisionNumber>=<eotPredictsAlt>;
 break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it|case <it>:}; separator="\n">
 {
 <targetState>
 }
 break;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
try
{
 alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
}
catch (NoViableAltException nvae)
{
 DebugRecognitionException(nvae);
 throw;
}
>>

/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated
 strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
private class DFA<dfa.decisionNumber> : DFA
{
 private const string DFA<dfa.decisionNumber>_eotS =
  "<dfa.javaCompressedEOT; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_eofS =
  "<dfa.javaCompressedEOF; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_minS =
  "<dfa.javaCompressedMin; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_maxS =
  "<dfa.javaCompressedMax; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_acceptS =
  "<dfa.javaCompressedAccept; wrap="\"+\n\t\t\"">";
 private const string DFA<dfa.decisionNumber>_specialS =
  "<dfa.javaCompressedSpecial; wrap="\"+\n\t\t\"">}>";
 private static readonly string[] DFA<dfa.decisionNumber>_transitionS
 =
  {
   <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
  };

 private static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
 private static readonly short[] DFA<dfa.decisionNumber>_eof =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
 private static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
 private static readonly char[] DFA<dfa.decisionNumber>_max =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
 private static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
 private static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
 private static readonly short[][] DFA<dfa.decisionNumber>_transition;

 static DFA<dfa.decisionNumber>()
 {
  int numStates = DFA<dfa.decisionNumber>_transitionS.Length;
  DFA<dfa.decisionNumber>_transition
 = new short[numStates][];
  for ( int i=0; i \< numStates; i++ )
  {
   DFA<dfa.decisionNumber>_transition[i] =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
  }
 }

 public DFA<dfa.decisionNumber>( BaseRecognizer recognizer<if(dfa.specialStateSTs)>,
SpecialStateTransitionHandler specialStateTransition<endif> )
<if(dfa.specialStateSTs)>
  : base(specialStateTransition)
```

```
<endif>
 {
  this.recognizer = recognizer;
  this.decisionNumber = <dfa.decisionNumber>;
  this.eot = DFA<dfa.decisionNumber>_eot;
  this.eof = DFA<dfa.decisionNumber>_eof;
  this.min = DFA<dfa.decisionNumber>_min;
  this.max = DFA<dfa.decisionNumber>_max;
  this.accept = DFA<dfa.decisionNumber>_accept;
  this.special = DFA<dfa.decisionNumber>_special;
  this.transition = DFA<dfa.decisionNumber>_transition;
 }

 public override string Description { get { return "<dfa.description>"; } }

 public override void Error(NoViableAltException nvae)
 {
  DebugRecognitionException(nvae);
 }
}<\n>
<if(dfa.specialStateSTs)>
private
 int SpecialStateTransition<dfa.decisionNumber>(DFA dfa, int s, IIntStream _input)<! throws
NoViableAltException!>
{
 <if(LEXER)>
 IIntStream input = _input;
 <endif>
 <if(PARSER)>
 ITokenStream input = (ITokenStream)_input;
 <endif>
 <if(TREE_PARSER)>
 ITreeNodeStream input = (ITreeNodeStream)_input;
 <endif>
 int _s = s;
 s = -1;
 <! pull these outside the switch cases to save space on locals !>
 int LA<dfa.decisionNumber>_1 = input.LA(1);
 int index<dfa.decisionNumber>_1 = input.Index;
 switch (_s)
 {
 <dfa.specialStateSTs:{state |case <i0>:<! compressed special state numbers 0..n-1 !>
 <state>}; separator="\n">

 default:
  break;
 }
 if (s >= 0)
  return s;

<if(backtracking)>
 if (state.backtracking > 0) {state.failed=true; return -1;}
<endif>
 NoViableAltException nvae = new NoViableAltException(dfa.Description, <dfa.decisionNumber>, _s, input);
 dfa.Error(nvae);
 throw nvae;
}
<endif>
>>

/** A state in a cyclic
 DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
<if(semPredState)>
 <! get next lookahead symbol to test edges, then rewind !>
 input.Rewind();
<endif>
 <edges; separator="\nelse ">
<if(semPredState)>
 <! return input cursor to state before we rewound !>
 input.Seek(index<decisionNumber>_1);
<endif>
 break;
}
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> && (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S
```

```
andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands)
 ::= "(<operands; separator=\"||\">)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "EvaluatePredicate(<pred>_fragment)"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<k>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <%
(LA<decisionNumber>_<k><ge()><lower> && LA<decisionNumber>_<k><le()><upper>)
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::=
"(input.LA(<k>)<ge()><lower> && input.LA(<k>)<le()><upper>)"

le() ::= "\<="
ge() ::= ">="

setTest(ranges) ::= <<
<ranges; separator="||">
>>

// A T T R I B U T E S

attributeScope(scope) ::= <<
<if(scope)>
<if(scope.attributes)>
protected sealed partial class
 <scope.name>_scope
{
 <scope.attributes:{it|public <it.decl>;}; separator="\n">

 public <scope.name>_scope(<grammar.recognizerName> grammar) { OnCreated(grammar); }
 partial void OnCreated(<grammar.recognizerName> grammar);
}
<if(scope.actions.scopeinit)>
protected void <scope.name>_scopeInit( <scope.name>_scope scope )
```

```
{
 <scope.actions.scopeinit>
}
<else>
partial void <scope.name>_scopeInit( <scope.name>_scope scope );
<endif>
<if(scope.actions.scopeafter)>
protected void <scope.name>_scopeAfter( <scope.name>_scope scope )
{
 <scope.actions.scopeafter>
}
<else>
partial void <scope.name>_scopeAfter( <scope.name>_scope scope );
<endif>
protected readonly ListStack\<<scope.name>_scope> <scope.name>_stack = new
ListStack\<<scope.name>_scope>();
<endif>
<endif>
>>

globalAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

ruleAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

returnStructName(it) ::= "<it.name>_return"

returnType(ruleDescriptor) ::= <%
<if(ruleDescriptor.returnScope.attributes
 && ruleDescriptor.hasMultipleReturnValues)>
 <ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<elseif(ruleDescriptor.hasMultipleReturnValues)>
 <ruleReturnBaseType()>
<elseif(ruleDescriptor.hasSingleReturnValue)>
 <ruleDescriptor.singleValueReturnType>
<else>
 void
<endif>
%>

/** Generate the C# type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
```

```
 <ruleReturnBaseType()>
<elseif(referencedRule.hasSingleReturnValue)>
 <referencedRule.singleValueReturnType>
<else>
 void
<endif>
%>

delegateName(it) ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
default(<typeName>)
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <%
<ruleLabelType(label.referencedRule)> <label.label.text>
 = <initValue(ruleLabelType(label.referencedRule))>;
%>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(scope.attributes && ruleDescriptor.hasMultipleReturnValues)>
<returnScopeModifier(grammar=grammar,ruleDescriptor=ruleDescriptor)> sealed partial class
<ruleDescriptor:returnStructName()> : <ruleReturnBaseType()><@ruleReturnInterfaces()>
{
 <scope.attributes:{it|public <it.decl>;}; separator="\n">
 <@ruleReturnMembers()>
}
<endif>
>>

ruleReturnBaseType() ::= <%
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope\<<labelType>>
%>

@returnScope.ruleReturnMembers() ::= <<
public <ruleDescriptor:returnStructName()>(<grammar.recognizerName> grammar) {OnCreated(grammar);}
partial void OnCreated(<grammar.recognizerName> grammar);
>>
```

```
parameterScope(scope) ::= <<
<scope.attributes:{it|<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr)
 ::= <<
<attr.name; format="id">
>>

parameterSetAttributeRef(attr,expr) ::= <<
<attr.name; format="id"> =<expr>;
>>

scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id">
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id">
<else>
<scope>_stack.Peek().<attr.name; format="id">
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <%
<if(negIndex)>
<scope>_stack[<scope>_stack.Count - <negIndex> - 1].<attr.name; format="id"> = <expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name; format="id"> = <expr>;
<else>
<scope>_stack.Peek().<attr.name; format="id"> = <expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.Count>0 && $function::name.Equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/**
 reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
```

```
(<scope>!=null?((<returnType(referencedRule)>)<scope>).<attr.name; format="id">:<initValue(attr.type)>)
<else>
<scope>
<endif>
%>

returnAttributeRef(ruleDescriptor,attr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id">
<else>
<attr.name; format="id">
<endif>
%>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <%
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name; format="id"> =<expr>;
<else>
<attr.name; format="id"> =<expr>;
<endif>
%>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.Text:default(string))"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.Type:0)"
tokenLabelPropertyRef_line(scope,attr)
 ::= "(<scope>!=null?<scope>.Line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.CharPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.Channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.TokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int.Parse(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.Start):default(<labelType>))"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.Stop):default(<labelType>))"
ruleLabelPropertyRef_tree(scope,attr) ::=
"(<scope>!=null?((<ASTLabelType>)<scope>.Tree):default(<ASTLabelType>))"
ruleLabelPropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
(<scope>!=null?(input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(<scope>.Start),
 input.TreeAdaptor.GetTokenStopIndex(<scope>.Start))):default(string))
<else>
(<scope>!=null?input.ToString(<scope>.Start,<scope>.Stop):default(string))
<endif>
%>

ruleLabelPropertyRef_st(scope,attr)
 ::= "(<scope>!=null?<scope>.Template:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(<scope>!=null?<scope>.Type:0)"

lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(<scope>!=null?<scope>.Line:0)"

lexerRuleLabelPropertyRef_pos(scope,attr) ::=
   "(<scope>!=null?<scope>.CharPositionInLine:-1)"

lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(<scope>!=null?<scope>.Channel:0)"

lexerRuleLabelPropertyRef_index(scope,attr) ::=
   "(<scope>!=null?<scope>.TokenIndex:0)"

lexerRuleLabelPropertyRef_text(scope,attr) ::=
   "(<scope>!=null?<scope>.Text:default(string))"

lexerRuleLabelPropertyRef_int(scope,attr) ::=
   "(<scope>!=null?int.Parse(<scope>.Text):0)"

// Somebody may ref
 $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.Start"
rulePropertyRef_stop(scope,attr) ::= "retval.Stop"
rulePropertyRef_tree(scope,attr) ::= "retval.Tree"
rulePropertyRef_text(scope,attr) ::= <%
<if(TREE_PARSER)>
input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start))
<else>
input.ToString(retval.Start,input.LT(-1))
<endif>
%>
```

```
rulePropertyRef_st(scope,attr) ::= "retval.Template"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr)
 ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.Tree = <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.Template =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <%
<if(backtracking)>
if (<actions.(actionScope).synpredgate>)<\n>
{<\n>
<@indentedAction()><\n>
}
<else>
<action>
<endif>
%>

@execAction.indentedAction() ::= <<
 <action>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static readonly BitSet <name> = new BitSet(new ulong[]{<words64:{it|<it>UL};separator=",">});
>>

codeFileExtension() ::= ".cs"

true_value() ::= "true"
false_value() ::= "false"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp3/CSharp3.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/

finishedBacktracking(block) ::= <<
<if(!ruleDescriptor.isSynPred)>
```

```
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
    <block>
<else>
<block>
<endif>
<endif>
>>

@ruleBody.setErrorReturnValue() ::= <<
retval.tree = self._adaptor.errorNode(self.input, retval.start, self.input.LT(-1), re)
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/**
 label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>
```

```
/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate
 that refactoring when
// I have more time.

matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <%
<super.matchSet(postmatchCode={<finishedBacktracking({self._adaptor.addChild(root_0,
<createNodeFromToken(...)>)})>}, ...)>
%>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={}) ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
<super.matchSet(postmatchCode={<finishedBacktracking({root_0 =
self._adaptor.becomeRoot(<createNodeFromToken(...)>, root_0)})>}, ...)>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
```

```
<finishedBacktracking({self._adaptor.addChild(root_0,
 <label>.tree)})>
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({root_0 = self._adaptor.becomeRoot(<label>.tree, root_0)})>
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(label, {<label>.tree})>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

// WILDCARD AST

wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
self._adaptor.addChild(root_0,
 <label>_tree)
})>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

createNodeFromToken(label,terminalOptions={}) ::= <%
<if(terminalOptions.node)>
<terminalOptions.node>(<label>) <! new MethodNode(IDLabel) !>
<else>
self._adaptor.createWithPayload(<label>)
<endif>
%>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
self._adaptor.setTokenBoundaries(retval.tree, retval.start, retval.stop)
})>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ASTParser.stg

No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Python by code generator.
*/
group ASTDbg;

astAccessor() ::= <<
def setTreeAdaptor(self, adaptor):
<if(grammar.grammarIsRoot)>
   self._adaptor = DebugTreeAdaptor(self.dbg, adaptor)
<else>
   self._adaptor = adaptor # delegator sends dbg adaptor
<endif>
   <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(self._adaptor)}>

def getTreeAdaptor(self):
    return self._adaptor

adaptor = property(getTreeAdaptor, setTreeAdaptor)<\n>
>>

createListenerAndHandshake() ::= <<
proxy = DebugEventSocketProxy(self,
adaptor=<if(TREE_PARSER)>self.input.getTreeAdaptor()<else>self._adaptor<endif>,
                            debug=debug_socket, port=port)
self.setDebugListener(proxy)
self.adaptor.setDebugListener(proxy)
self.input.setDebugListener(proxy)
#self.set<inputStreamType>(Debug<inputStreamType>(self.input, proxy))
proxy.handshake()
>>

@rewriteElement.pregen() ::= "self._dbg.location(<e.line>, <e.pos>)"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

```
*/

/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/

finishedBacktracking(block) ::= <<
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
   <block>
<else>
<block>
<endif>
>>

/**
```

```
 Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_first_0 = None
_last = None<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<finishedBacktracking({
<if(rewriteMode)>
retval.tree = _first_0
if self._adaptor.getParent(retval.tree) is not None and self._adaptor.isNil(self._adaptor.getParent(retval.tree)):
   retval.tree = self._adaptor.getParent(retval.tree)
<endif>
})>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = self.input.LT(1)
_save_last_<treeLevel> = _last
_first_<treeLevel> = None
<if(!rewriteMode)>
root_<treeLevel> = self._adaptor.nil()<\n>
<endif>
<root:element()>
<if(rewriteMode)>
<finishedBacktracking({
<if(root.el.rule)>
if _first_<enclosingTreeLevel>
 is None:
   _first_<enclosingTreeLevel> = <root.el.label>.tree<\n>
<else>
if _first_<enclosingTreeLevel> is None:
   _first_<enclosingTreeLevel> = <root.el.label><\n>
<endif>
})>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if self.input.LA(1) == DOWN:
   self.match(self.input, DOWN, None)
   <children:element()>
   self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)<\n>
<endif>
<if(!rewriteMode)>
self._adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>)<\n>
<endif>
_last = _save_last_<treeLevel>

>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree
 = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label><\n>
})>
<endif>
>>

/** label+=TOKEN auto construct */
```

```
tokenRefAndListLabel(token,label,elementIndex,terminalOptions={}) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,terminalOptions={})
 ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.wildcard(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<label>_tree = self._adaptor.dupTree(<label>)
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label>
})>
<endif>
>>

// SET AST
matchSet(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
_last = self.input.LT(1)
```

```
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<endif>
}, ...)>
>>

matchRuleBlockSet(s,label,elementIndex,postmatchCode,treeLevel="0",terminalOptions={})
 ::= <<
<matchSet(...)>
<noRewrite(...)> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode,terminalOptions={}) ::= <<
_last = self.input.LT(1)
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,elementIndex,debug,terminalOptions={}) ::= <<
<super.matchSet(postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(terminalOptions.node)>
<label>_tree = <terminalOptions.node>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
}, ...)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRef(...)>
<finishedBacktracking({
<if(!rewriteMode)>
self._adaptor.addChild(root_<treeLevel>, <label>.tree)
<else> <! rewrite mode !>
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label>.tree<\n>
<endif>
})>
>>

/** x+=rule auto construct
 */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(label, {<label>.tree})>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
root_<treeLevel> = self._adaptor.becomeRoot(<label>.tree, root_<treeLevel>)
})>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(label, {<label>.tree})>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope)
 ::= <<
_last = self.input.LT(1)
```

```
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,scope,terminalOptions={}) ::= <<
<if(terminalOptions.node)>
<terminalOptions.node>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
})>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Python/ASTTreeParser.stg

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2007-2008 Johannes Luber
* Copyright (c) 2005-2007 Kunle Odutola
* Copyright (c) 2005 Terence Parr
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
```

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/

parserMembers() ::= <<
 protected DebugTreeAdaptor adaptor;

 public ITreeAdaptor TreeAdaptor
 {
  get
  {
   return adaptor;
  }
  set
  {
<if(grammar.grammarIsRoot)>
   this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
   this.adaptor
 = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
   <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
  }
 }<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
```

```
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port,
<if(TREE_PARSER)>input.TreeAdaptor<else>adaptor<endif> );
DebugListener = proxy;
<inputStreamType> = new Debug<inputStreamType>( input, proxy );
try
{
 proxy.Handshake();
}
catch ( IOException ioe )
{
 ReportError( ioe );
}
>>

@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

@ctorForProfilingRootGrammar.finally() ::=<<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
>>

@ctorForPredefinedListener.superClassRef() ::= ": base( input, dbg )"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)><! don't create new
 adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>

//@rewriteElement.pregen() ::= "dbg.Location( <e.line>, <e.pos> );"
```

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2005-2009 Gokulakannan Somasundaram,

All rights reserved.

```
/*
* This code generating template and the associated Cpp runtime was produced by:
* Gokulakannan Somasundaram ( heavy lifting from C Run-time by Jim Idle )
*/
cTypeInitMap ::= [
 "int"      : "0",              // Integers     start out being 0
 "long"      : "0",              // Longs        start out being 0
 "float"      : "0.0",           // Floats       start out being 0
 "double"     : "0.0",           // Doubles      start out being 0
 "bool"         : "false",
   // Booleans     start out being Antlr C for false
 "byte"      : "0",              // Bytes        start out being 0
 "short"      : "0",              // Shorts       start out being 0
 "char"      : "0"              // Chars        start out being 0
]

leadIn(type) ::=
<<
/** \file
*  This <type> file was generated by $ANTLR version <ANTLRVersion>
*
*     -  From the grammar source file : <fileName>
*     -                            On : <generatedTimestamp>
<if(LEXER)>
```

```
*     -                 for the lexer : <name>Lexer
<endif>
<if(PARSER)>
*     -               for the parser : <name>Parser
<endif>
<if(TREE_PARSER)>
*     -           for the tree parser : <name>TreeParser
<endif>
*
* Editing it, at least manually, is not wise.
*
* C++ language generator and runtime by Gokulakannan Somasundaram ( heavy lifting from C Run-time by Jim Idle )
*
*
>>

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
 */
outputFile( LEXER,
        PARSER,
        TREE_PARSER,
        actionScope,
        actions,
        docComment,
        recognizer,
        name,
        tokens,
        tokenNames,
        rules,
        cyclicDFAs,
        bitsets,
        buildTemplate,
        buildAST,
        rewriteMode,
        profile,
        backtracking,
        synpreds,
        memoize,
        numRules,
        fileName,
        ANTLRVersion,
        generatedTimestamp,
        trace,
        scopes,
        superClass,
        literals
```

```
          ) ::=
<<
<leadIn("C++ source")>
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Gokulakannan Somasundaram, ElectronDB
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain
 the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY,
 WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<if(actions.(actionScope).header)>

/* =============================================================================
* This is what the grammar programmer asked us to put at the top of every file.
*/
<actions.(actionScope).header>
/* End of Header action.
* =============================================================================
*/
<endif>

/* ----------------------------------------
* Include the ANTLR3 generated header file.
*/
#include    "<name>.hpp"
```

```
<if(trace)>
#include \<iostream>
<endif>
<if(recognizer.grammar.delegators)>
// Include delegator definition header files
//
<recognizer.grammar.delegators: {g|#include "<g.recognizerName>.hpp" }; separator="\n">
<endif>

<actions.(actionScope).postinclude>
/* ----------------------------------------- */

<docComment>

<if(literals)>

<beginNamespace(actions)>

/**
 String literals used by <name> that we must do things like MATCHS() with.
*  C will normally just lay down 8 bit characters, and you can use L"xxx" to
*  get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
*  we perform this little trick of defining the literals as arrays of UINT32
*  and passing in the address of these.
*/
<literals:{it | static ANTLR_UCHAR lit_<i>[]  = <it>;}; separator="\n">

<endNamespace(actions)>

<endif>

/* ============================================================================= */

/* =============================================================================
* Start of recognizer
*/

<recognizer>

/* End of code
* =============================================================================
*/

>>
headerFileExtension() ::= ".hpp"

beginNamespace(actions) ::= <%
 <if(actions.(actionScope).namespace)>
 namespace <actions.(actionScope).namespace> {
 <endif>
%>

endNamespace(actions) ::=
 <%
 <if(actions.(actionScope).namespace)>
 }
 <endif>
%>



headerFile( LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,
            rewriteMode,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,
            generatedTimestamp,
            trace,
            scopes,
            superClass,
            literals
        ) ::=
<<
<leadIn("C++ header")>
<if(PARSER)>
* The parser <mainName()> has the callable functions (rules) shown below,
<endif>
<if(LEXER)>
* The lexer <mainName()> has the callable functions (rules) shown below,
<endif>
<if(TREE_PARSER)>
* The tree parser <mainName()> has the callable functions
 (rules) shown below,
<endif>
* which will invoke the code for the associated rule in the source grammar
* assuming that the input stream is pointing to a token/text stream that could begin
* this rule.
*
* For instance if you call the first (topmost) rule in a parser grammar, you will
* get the results of a full parse, but calling a rule half way through the grammar will
* allow you to pass part of a full token stream to the parser, such as for syntax checking
* in editors and so on.
*
*/
// [The "BSD license"]
// Copyright (c) 2005-2009 Gokulakannan Somasundaram.
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list
 of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 SUCH DAMAGE.

#ifndef _<name>_H
#define _<name>_H
<actions.(actionScope).preincludes>
/* =============================================================================
* Standard antlr3 C++ runtime definitions
*/
#include \<antlr3.hpp>

/* End of standard antlr 3 runtime definitions
* =============================================================================
*/

<actions.(actionScope).includes>

<if(recognizer.grammar.delegates)>
// Include delegate definition header files
//
<recognizer.grammar.delegates: {g|#include "<g.recognizerName>.hpp"}; separator="\n">

<endif>


<actions.(actionScope).header>

#ifdef WIN32
// Disable: Unreferenced parameter,       - Rules with parameters that are not used
//          constant conditional,       - ANTLR realizes that a prediction is always true (synpred usually)
//          initialized but unused variable     - tree rewrite variables declared but not needed
//          Unreferenced local variable      - lexer rule declares but does not always
 use _type
//          potentially unitialized variable used   - retval always returned from a rule
//   unreferenced local function has been removed - susually getTokenNames or freeScope, they can go without
warnigns
//
// These are only really displayed at warning level /W4 but that is the code ideal I am aiming at
// and the codegen must generate some of these warnings by necessity, apart from 4100, which is
// usually generated when a parser rule is given a parameter that it does not use. Mostly though
// this is a matter of orthogonality hence I disable that one.
//
#pragma warning( disable : 4100 )
#pragma warning( disable : 4101 )
#pragma warning( disable : 4127 )
#pragma warning( disable : 4189 )
#pragma warning( disable : 4505 )
#pragma warning( disable : 4701 )
#endif
<if(backtracking)>

/* ========================
* BACKTRACKING IS ENABLED
* ========================
*/
```

```
<endif>

<beginNamespace(actions)>

<if(recognizer.grammar.delegators)>
// Include delegator definition
 classes
//
<recognizer.grammar.delegators: {g|class <g.recognizerName>; }; separator="\n">
<endif>

<actions.(actionScope).traits>
typedef <name>Traits <name>ImplTraits;

<rules:{r | <if(r.ruleDescriptor.isSynPred)> struct <r.ruleDescriptor.name> {\}; <endif>}; separator="\n">

class <name>Tokens
{
public:
 /** Symbolic definitions of all the tokens that the <grammarType()> will work with.
  *
  * Antlr will define EOF, but we can't use that as it it is too common in
  * in C header files and that would be confusing. There is no way to filter this out at the moment
  * so we just undef it here for now. That isn't the value we get back from C recognizers
  * anyway. We are looking for ANTLR_TOKEN_EOF.
  */
 enum Tokens
 {
  EOF_TOKEN = <name>ImplTraits::CommonTokenType::TOKEN_EOF
  <tokens:{it | , <it.name> = <it.type> }; separator="\n">
 };

};

/** Context tracking structure for <mainName()>
*/
class <name> : public <componentBaseType()>, public <name>Tokens
{
public:
 typedef
 <name>ImplTraits ImplTraits;
 typedef <name> ComponentType;
 typedef ComponentType::StreamType StreamType;
 typedef <componentBaseType()> BaseType;
 typedef ImplTraits::RecognizerSharedStateType\<StreamType> RecognizerSharedStateType;
 typedef StreamType InputType;
<if(recognizer.filterMode)>
 static const bool IsFiltered = true;
```

```
<else>
 static const bool IsFiltered = false;
<endif>

 <scopes:{it | <if(it.isDynamicGlobalScope)><globalAttributeScopeDecl(it)><endif>}>
 <rules:{r | <if(r.ruleDescriptor.ruleScope)><ruleAttributeScopeDecl(scope=r.ruleDescriptor.ruleScope)><endif>}>

private:
<if(recognizer.grammar.delegates)>
 <recognizer.grammar.delegates:
        {g|<g.recognizerName>*  m_<g:delegateName()>;}; separator="\n">
<endif>
<if(recognizer.grammar.delegators)>
 <recognizer.grammar.delegators:
        {g|<g.recognizerName>*  m_<g:delegateName()>;}; separator="\n">
<endif>
<scopes:{it | <if(it.isDynamicGlobalScope)>
   <globalAttributeScopeDef(it)>
<endif>}; separator="\n\n">
<rules:
 {r |<if(r.ruleDescriptor.ruleScope)>
   <ruleAttributeScopeDef(scope=r.ruleDescriptor.ruleScope)>
<endif>}>
   <@members>
   <@end>

public:
   <name>(InputType* instream<recognizer.grammar.delegators:{g|, <g.recognizerName>*
<g:delegateName()>}>);
   <name>(InputType* instream, RecognizerSharedStateType* state<recognizer.grammar.delegators:{g|,
<g.recognizerName>* <g:delegateName()>}>);

   void init(InputType* instream <recognizer.grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()>}>
);

   <actions.(actionScope).context>

<if(LEXER)>
<if(recognizer.filterMode)>
   void  memoize(ANTLR_MARKER  ruleIndex, ANTLR_MARKER  ruleParseStart);
   bool  alreadyParsedRule(ANTLR_MARKER ruleIndex);
   <filteringNextToken()>
<endif>
   <rules:{r | <if(!r.ruleDescriptor.isSynPred)><headerReturnType(ruleDescriptor=r.ruleDescriptor)>
m<r.ruleDescriptor.name>( <r.ruleDescriptor.parameterScope:parameterScope()>);<endif>}; separator="\n">
   <rules:{r | <if(r.ruleDescriptor.isSynPred)>
  <headerReturnType(ruleDescriptor=r.ruleDescriptor)> msynpred( antlr3::ClassForwarder\<
<r.ruleDescriptor.name> > <r.ruleDescriptor.parameterScope:parameterScope()>);
```

```
   void m<r.ruleDescriptor.name>_fragment (<r.ruleDescriptor.parameterScope:parameterScope()>);<endif>};
separator="\n">
<endif>
<if(!LEXER)>
   <rules:{r | <headerReturnScope(ruleDescriptor=r.ruleDescriptor)>}>
   <rules:{r | <if(!r.ruleDescriptor.isSynPred)> <headerReturnType(ruleDescriptor=r.ruleDescriptor)>
<r.ruleDescriptor.name> (<r.ruleDescriptor.parameterScope:parameterScope()>); <endif>}; separator="\n">
   <rules:{r | <if(r.ruleDescriptor.isSynPred)>  <headerReturnType(ruleDescriptor=r.ruleDescriptor)> msynpred(
antlr3::ClassForwarder\< <r.ruleDescriptor.name> > <r.ruleDescriptor.parameterScope:parameterScope()>);
   void m<r.ruleDescriptor.name>_fragment (<r.ruleDescriptor.parameterScope:parameterScope()>);<endif>};
separator="\n">
<! generate rule/method definitions for imported rules
 so they
  appear to be defined in this recognizer. !>
   // Delegated rules
<recognizer.grammar.delegatedRules:{ruleDescriptor|
   <headerReturnType(ruleDescriptor)>
<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>);}; separator="\n">
<endif>

   const char *    getGrammarFileName();
   void            reset();
   ~<name>();

};

// Function protoypes for the constructor functions that external translation units
// such as delegators and delegates may wish to call.
//
<if(!recognizer.grammar.grammarIsRoot)>
extern ANTLR_UINT8*   <recognizer.grammar.composite.rootGrammar.recognizerName>TokenNames[];
<endif>


/* End of token definitions for <name>
* ==============================================================================
*/

<endNamespace(actions)>

#endif

/* END - Note:Keep extra line feed to satisfy UNIX systems */

>>

grammarType() ::= <%
<if(PARSER)>
```

```
parser
<endif>
<if(LEXER)>
lexer
<endif>
<if(TREE_PARSER)>
tree parser
<endif>
%>

componentType()
 ::= <<
<if(PARSER)>
<name>ImplTraits::ParserType
<endif>
<if(LEXER)>
<name>ImplTraits::LexerType
<endif>
<if(TREE_PARSER)>
<name>ImplTraits::TreeParserType
<endif>
>>

componentBaseType() ::= <%
<if(PARSER)>
<name>ImplTraits::BaseParserType
<endif>
<if(LEXER)>
<name>ImplTraits::BaseLexerType
<endif>
<if(TREE_PARSER)>
<name>ImplTraits::BaseTreeParserType
<endif>
%>

streamType() ::= <<
<if(PARSER)>
<name>ImplTraits::ParserType::StreamType
<endif>
<if(LEXER)>
<name>ImplTraits::LexerType::StreamType
<endif>
<if(TREE_PARSER)>
<name>ImplTraits::TreeParserType::StreamType
<endif>
>>
```

```
mainName() ::= <%
<if(PARSER)>
<name>
<endif>
<if(LEXER)>
<name>
<endif>
<if(TREE_PARSER)>
<name>
<endif>
%>

headerReturnScope(ruleDescriptor) ::= "<returnScope(scope=ruleDescriptor.returnScope)>"

headerReturnType(ruleDescriptor) ::= <%
<if(LEXER)>
<if(!ruleDescriptor.isSynPred)>
void
<else>
<returnType()>
<endif>
<else>
<returnType()>
<endif>
%>

// Produce the lexer output
//
lexer(  grammar,
 name,
        tokens,
       scopes,
       rules,
       numRules,
       filterMode,
       superClass,
       labelType="ImplTraits::CommonTokenType*") ::= <<

using namespace antlr3;

<beginNamespace(actions)>

<if(filterMode)>

/* Override the normal MEMOIZE and HAVEALREADYPARSED macros as this is a filtering
* lexer. In filter mode, the memoizing and backtracking are gated at BACKTRACKING > 1 rather
* than just BACKTRACKING. In some cases this might generate code akin to:
*   if (BACKTRACKING) if (BACKTRACKING > 1) memoize.
```

```
*/
void <name>::memoize(ANTLR_MARKER  ruleIndex, ANTLR_MARKER  ruleParseStart)
{
 BaseType* base = this;
 if ( this->get_backtracking()>1 )
  base->memoize( ruleIndex, ruleParseStart );

}

bool <name>::alreadyParsedRule(ANTLR_MARKER ruleIndex)
{
 BaseType* base = this;
 if ( this->get_backtracking() > 1 )
  return base->haveParsedRule(ruleIndex);
 return false;
}

<endif>

/* =============================================================================
* Lexer
 matching rules end.
* =============================================================================
*/

<scopes:{it |<if(it.isDynamicGlobalScope)><globalAttributeScope(it)><endif>}>

<actions.lexer.members>

<name>::~<name>()
{
<if(memoize)>
 RuleMemoType* rulememo = this->getRuleMemo();
 if(rulememo != NULL)
 {
  delete rulememo;
  this->setRuleMemo(NULL);
 }
<endif>
<if(grammar.directDelegates)>
 // Free the lexers that we delegated to
 // functions to. NULL the state so we only free it once.
 //
 <grammar.directDelegates:
        {g| m_<g:delegateName()>->set_lexstate(NULL);
        delete m_<g:delegateName()>; }; separator="\n">
<endif>
}
```

```
void
<name>::reset()
{
   this->get_rec()->reset();
}

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
const char* <name>::getGrammarFileName()
{
 return fileName;
}

/** \brief Create a new lexer called
 <name>
*
* \param[in]    instream Pointer to an initialized input stream
* \return
*     - Success p<name> initialized for the lex start
*     - Fail NULL
*/
<name>::<name>(StreamType* instream<grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()>}>)
:<name>ImplTraits::BaseLexerType(ANTLR_SIZE_HINT, instream, NULL)
{
 // See if we can create a new lexer with the standard constructor
 //
 this->init(instream <grammar.delegators:{g|, <g:delegateName()>}>);
}

/** \brief Create a new lexer called <name>
*
* \param[in]    instream Pointer to an initialized input stream
* \param[state] state Previously created shared recognizer stat
* \return
*     - Success p<name> initialized for the lex start
*     - Fail NULL
*/
<name>::<name>(StreamType* instream, RecognizerSharedStateType* state<grammar.delegators:{g|,
<g.recognizerName>* <g:delegateName()>}>)
:<name>ImplTraits::BaseLexerType(ANTLR_SIZE_HINT, instream, state)
{
 this->init(instream <grammar.delegators:{g|, <g:delegateName()>}
 >);
}

void <name>::init(StreamType* instream<grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()>} >)
{
   /* -------------------------------------------------------------------
    * Memory for basic structure is allocated, now to fill in
    * in base ANTLR3 structures. We initialize the function pointers
    * for the standard ANTLR3 lexer function set, but upon return
    * from here, the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here so you can get a sense
    * of what goes where.
    */

<if(memoize)>
<if(grammar.grammarIsRoot)>
   // Create a LIST for recording rule memos.
   //
   this->setRuleMemo( new IntTrie(15) ); /* 16 bit depth is enough for 32768 rules! */
<endif>
<endif>

<if(grammar.directDelegates)>
 // Initialize the lexers that we are going to delegate some
 // functions to.
 //
 <grammar.directDelegates:
        {g|m_<g:delegateName()>
 = new <g.recognizerName>(instream, this->get_lexstate(), this<grammar.delegators:{g|, <g:delegateName()>}>);};
separator="\n">
<endif>
<if(grammar.delegators)>
 // Install the pointers back to lexers that will delegate us to perform certain functions
 // for them.
 //
 <grammar.delegators:
        {g| m_<g:delegateName()> = <g:delegateName()>;}; separator="\n">
<endif>
}

<if(cyclicDFAs)>

/* =========================================================================
* DFA tables for the lexer
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
```

```
/* =============================================================================
* End of DFA tables for the lexer
*/
<endif>

/* =============================================================================
* Functions to match the lexer grammar defined tokens from the input stream
*/

<rules; separator="\n\n">

/* =============================================================================
* Lexer matching rules end.
* =============================================================================
 */
<if(synpreds)>

/* =============================================================================
* Lexer syntactic predicates
*/
<synpreds:{p | <lexerSynpred(predname=p)>}>
/* =============================================================================
* Lexer syntactic predicates end.
* =============================================================================
*/
<endif>

/* End of Lexer code
* ================================================
* ================================================
*/

<endNamespace(actions)>

>>


filteringNextToken() ::= <<
<name>ImplTraits::CommonTokenType*
<name>ImplTraits::TokenSourceType::nextToken()
{
    LexerType*   lexer;
    typename LexerType::RecognizerSharedStateType* state;

    lexer   = this->get_super();
    state = lexer->get_lexstate();

    /* Get rid of any previous token (token factory takes care of
     * any deallocation when this token is finally used up.

 */
    state->set_token_present( false );
    state->set_error( false );     /* Start out without an exception */
    state->set_failedflag(false);

    /* Record the start of the token in our input stream.
     */
    state->set_tokenStartCharIndex( lexer->index();
    state->set_tokenStartCharPositionInLine( lexer->getCharPositionInLine() );
    state->set_tokenStartLine( lexer->getLine() );
    state->set_text("");

    /* Now call the matching rules and see if we can generate a new token
     */
    for (;;)
    {
 if (lexer->LA(1) == ANTLR_CHARSTREAM_EOF)
 {
    /* Reached the end of the stream, nothing more to do.
     */
    CommonTokenType&    teof = m_eofToken;

    teof.set_startIndex(lexer->getCharIndex());
    teof.set_stopIndex(lexer->getCharIndex());
    teof.setLine(lexer->getLine());
    return  &teof;
 }

 state->set_token_present(false);
 state->set_error(false);     /* Start out without an exception */

 {
    ANTLR_MARKER   m;

    m = this->get_istream()->mark();
    state->set_backtracking(1);    /*
 No exceptions */
    state->set_failedflag(false);

    /* Call the generated lexer, see if it can get a new token together.
     */
    lexer->mTokens();
       state->set_backtracking(0);

      <! mTokens backtracks with synpred at BACKTRACKING==2
     and we set the synpredgate to allow actions at level 1. !>
    if(state->get_failed())
    {
     lexer->rewind(m);
     lexer->consume(); <! advance one char and try again !>
    }
    else
    {
     lexer->emit();     /* Assemble the token and emit it to the stream */
     TokenType& tok = state->get_token();
     return &tok;
    }
 }
    }
 }
 >>

 actionGate() ::= "this->get_backtracking()==0"

 filteringActionGate() ::= "this->get_backtracking()==1"

 /** How to generate a parser */
 genericParser(  grammar, name, scopes, tokens, tokenNames, rules, numRules,
               bitsets, inputStreamType, superClass,
               labelType, members, rewriteElementType,
               filterMode, ASTLabelType="ImplTraits::TreeType*")
 ::= <<

 using namespace antlr3;
 <if(grammar.grammarIsRoot)>
 /** \brief Table of all token names in symbolic order, mainly used for
 *         error reporting.
 */
 ANTLR_UINT8* <name>TokenNames[<length(tokenNames)>+4]
     = {
        (ANTLR_UINT8*) "\<invalid>",       /* String to print to indicate an invalid token */
        (ANTLR_UINT8*) "\<EOR>",
        (ANTLR_UINT8*) "\<DOWN>",
        (ANTLR_UINT8*) "\<UP>",
        <tokenNames:{it |(ANTLR_UINT8*) <it>}; separator=",\n">
       };
 <endif>

    <@members>

    <@end>

 /** \brief Name of the grammar file that generated this code
```

```
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
const char* <name>::getGrammarFileName()
{
 return fileName;
}
/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
<name>::<name>(
 StreamType* instream<grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()>}>)
<constructorInitializerType("NULL")>
{
 // See if we can create a new parser with the standard constructor
 //
 this->init(instream<grammar.delegators:{g|, <g:delegateName()>}>);
}

/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
<name>::<name>( StreamType* instream, RecognizerSharedStateType* state<grammar.delegators:{g|,
<g.recognizerName>* <g:delegateName()>}>)
<constructorInitializerType("state")>
{
 this->init(instream <grammar.delegators:{g|, <g:delegateName()>}>);
}

void <name>::init(StreamType* instream<grammar.delegators:{g|, <g.recognizerName>* <g:delegateName()>}>)
{
   <actions.parser.apifuncs>
<if(memoize)>
<if(grammar.grammarIsRoot)>
   /* Create a LIST for recording rule memos.
    */
    typedef RecognizerSharedStateType::RuleMemoType
 RuleMemoType;
    this->setRuleMemo( new RuleMemoType(15) ); /* 16 bit depth is enough for 32768 rules! */<\n>
<endif>
```

```
<endif>
<if(grammar.directDelegates)>
 // Initialize the lexers that we are going to delegate some
 // functions to.
 //
 <grammar.directDelegates:
        {g|m_<g:delegateName()> = new <g.recognizerName>(instream, this->get_psrstate(),
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
 <endif>
       <if(grammar.delegators)>
     // Install the pointers back to lexers that will delegate us to perform certain functions
     // for them.
     //
      <grammar.delegators: {g| m_<g:delegateName()> = <g:delegateName()>;}; separator="\n">
 <endif>
 /* Install the token table
 */
 this->get_psrstate()->set_tokenNames( <grammar.composite.rootGrammar.recognizerName>TokenNames );

 <@debugStuff()>

}

void
<name>::reset()
{
   this->get_rec()->reset();
}

/** Free the parser resources
*/
<name>::~<name>()
{
   <@cleanup>
   <@end>
<if(grammar.directDelegates)>
 //
 Free the parsers that we delegated to
 // functions to.NULL the state so we only free it once.
 //
 <grammar.directDelegates:
        {g| m_<g:delegateName()>->set_psrstate( NULL );
        delete m_<g:delegateName()>;}; separator="\n">
<endif>
<if(memoize)>
<if(grammar.grammarIsRoot)>
 if(this->getRuleMemo() != NULL)
 {
  delete this->getRuleMemo();
  this->setRuleMemo(NULL);
 }
<endif>
<endif>
}

/** Return token names used by this <grammarType()>
*
* The returned pointer is used as an index into the token names table (using the token
* number as the index).
*
* \return Pointer to first char * in the table.
*/
static ANTLR_UINT8** getTokenNames()
{
       return <grammar.composite.rootGrammar.recognizerName>TokenNames;
}

   <members>

/* Declare the bitsets
*/
<bitsets:{it | <bitsetDeclare(bitsetname={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                 words64=it.bits, traits={<name>ImplTraits} )>}>


<if(cyclicDFAs)>

/* =========================================================================
 * DFA tables for the parser
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =========================================================================
* End of DFA tables for the parser
*/
<endif>

/* ==============================================
* Parsing rules
*/
<rules; separator="\n\n">
<if(grammar.delegatedRules)>
 // Delegated methods that appear to be a part of this
 // parser
 //
<grammar.delegatedRules:{ruleDescriptor|
   <returnType()> <name>::<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope()>)
   {
       <if(ruleDescriptor.hasReturnValue)>return <endif>m_<ruleDescriptor.grammar:delegateName()>-><ruleDescriptor.name>(<if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", "><endif>);
   \}}; separator="\n">

<endif>
/* End of parsing rules
* ==============================================
*/

/* ==============================================
 * Syntactic predicates
*/
<synpreds:{p | <synpred(predname=p)>}>
/* End of syntactic predicates
* ==============================================
*/

>>

constructorInitializerType(rec_state) ::=<<
<if(PARSER)>
   :ImplTraits::BaseParserType(ANTLR_SIZE_HINT, instream, <rec_state>)
<endif>
<if(TREE_PARSER)>
   :ImplTraits::BaseTreeParserType(ANTLR_SIZE_HINT, instream, <rec_state>)
<endif>
>>

parser( grammar,
  name,
  scopes,
  tokens,
  tokenNames,
  rules,
  numRules,
  bitsets,
  ASTLabelType,
  superClass="Parser",
  labelType="ImplTraits::CommonTokenType*",
  members={<actions.parser.members>}
  ) ::= <<
<beginNamespace(actions)>
<genericParser(inputStreamType="CommonTokenStreamType*", rewriteElementType="Token", filterMode=false, ...)>
<endNamespace(actions)>
>>
```

```
/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser( grammar,
   name,
   scopes,
   tokens,
   tokenNames,
   globalAction,
   rules,
   numRules,
   bitsets,
   filterMode,
   labelType={<ASTLabelType>},
   ASTLabelType="ImplTraits::TreeType*",
   superClass="TreeParser",
   members={<actions.treeparser.members>}
   )
 ::= <<
<beginNamespace(actions)>
<genericParser(inputStreamType="CommonTreeNodeStream*", rewriteElementType="Node", ...)>
<endNamespace(actions)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
void <name>::m<ruleName>_fragment( <ruleDescriptor.parameterScope:parameterScope()> )
{
 <ruleLabelDefs()>
 <ruleLabelInitializations()>
<if(trace)>
    ANTLR_PRINTF("enter <ruleName> %d failed = %d, backtracking = %d\\n", this->LT(1),failed,this-
>get_backtracking() );
   <block>
   ANTLR_PRINTF("exit <ruleName> %d, failed = %d, backtracking = %d\\n", this->LT(1),failed,this-
>get_backtracking());

<else>
   <block>
```

```
<endif>

goto rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex: ;
}
// $ANTLR end <ruleName>
>>

synpred(predname) ::= <<

bool <name>::msynpred( antlr3::ClassForwarder\< <predname> >  )
{
   ANTLR_MARKER   start;
   bool  success;

   this->inc_backtracking();
   <@start()>
   start = this->mark();
   this->m<predname>_fragment();     // can never throw exception
   success = !( this->get_failedflag() );
   this->rewind(start);
   <@stop()>
   this->dec_backtracking();
   this->set_failedflag(false);
   return success;
}<\n>
>>

lexerSynpred(predname) ::= <<
<synpred(predname)>
>>

ruleMemoization(rname) ::= <<
<if(memoize)>
if ( (this->get_backtracking()>0) && (this->haveParsedRule(<ruleDescriptor.index>))
 )
{
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
 retval.start = 0;<\n>
<endif>
<endif>
   <(ruleDescriptor.actions.after):execAfter()>
   <finalCode(finalBlock=finally)>
<if(!ruleDescriptor.isSynPred)>
   <scopeClean()><\n>
<endif>
   return <ruleReturnValue()>;
```

```
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
if  (this->hasException())
{
   goto rule<ruleDescriptor.name>Ex;
}
<if(backtracking)>
if (this->hasFailed())
{
   <scopeClean()>
   <@debugClean()>
   return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if (this->get_backtracking()>0)
{
   this->set_failedflag( true );
   <scopeClean()>
   return <ruleReturnValue()>;
}
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple
 return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
/**
* $ANTLR start <ruleName>
* <fileName>:<description>
*/
<returnType()>
<name>::<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>)
{
   <if(trace)>ANTLR_PRINTF("enter <ruleName> %s failed=%d, backtracking=%d\n", this->LT(1), this-
>get_backtracking() );<endif>
   <ruleDeclarations()>
   <ruleDescriptor.actions.declarations>
   <ruleLabelDefs()>
   <ruleInitializations()>
   <ruleDescriptor.actions.init>
   <ruleMemoization(rname=ruleName)>
   <ruleLabelInitializations()>

   <if(actions.(actionScope).rulecatch)>
    try {
   <else>
   <if(exceptions)>
      try {
   <endif>
   <endif>
   <@preamble()>
   {
      <block>
   }
   <ruleCleanUp()>

<if(exceptions)>
   <(ruleDescriptor.actions.after):execAfter()>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
   <if(!emptyRule)>
    if (this->hasException())
        {
           this->preporterror();
           this->precover();
           <@setErrorReturnValue()>
        }
        <if(ruleDescriptor.actions.after)>
        else
        {
           <(ruleDescriptor.actions.after):execAfter()>
        }<\n>
        <endif>
 <if(actions.(actionScope).rulecatch)>
     } <actions.(actionScope).rulecatch>
        <endif>
   <endif>
<endif>

   <if(trace)>ANTLR_PRINTF("exit <ruleName> %d failed=%s backtracking=%s\n", this->LT(1), failed, this->get_backtracking() );<endif>
   <memoize()>
<if(finally)>
   <finalCode(finalBlock=finally)>
<endif>
```

```
   <scopeClean()>
   <@postamble()>
   return <ruleReturnValue()>;
}
/* $ANTLR end <ruleName> */
>>

finalCode(finalBlock) ::= <<
{
   <finalBlock>
}

>>

catch(decl,action) ::= <<
/* catch(decl,action)
*/
}catch (<e.decl>) {
  <e.action>
}
>>

ruleDeclarations() ::= <<

<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval(this);<\n>
<else>
<if(PARSER)>
   <name>ImplTraits::RuleReturnValueType
 _antlr_rule_exit(this);
<endif>
<if(ruleDescriptor.returnScope)>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name>;
}>
<endif>
<endif>
<if(memoize)>
ANTLR_MARKER <ruleDescriptor.name>_StartIndex;
<endif>
>>

ruleInitializations() ::= <<
/* Initialize rule variables
*/
<if(ruleDescriptor.returnScope)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.returnScope.attributes:{ a | <if(a.initValue)>retval.<a.name> = <a.initValue>;<endif> }>
```

```
<else>
<ruleDescriptor.returnScope.attributes:{ a | <if(a.initValue)><a.name> = <a.initValue>;<endif> }>
<endif>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex = this->index();<\n>
<endif>
<ruleDescriptor.useScopes:{it | m_<it>_stack.push(<it>Scope()); }; separator="\n">
<ruleDescriptor.ruleScope:{it | m_<it.name>_stack.push(<it.name>Scope()); }; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
    :{it |<labelType> <it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{it |ImplTraits::TokenPtrsListType list_<it.label.text>;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(); separator="\n">
>>

ruleLabelInitializations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
retval.call_start_placeholder();
<endif>
<endif>
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |<labelType> <it.label.text> = NULL;}; separator="\n"
>
<ruleDescriptor.charLabels:{it |ANTLR_UINT32 <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{it | ImplTraits::IntTrieType<CommonTokenType>* list_<it.label.text>;}; separator="\n"
>
>>

lexerRuleLabelInit() ::= <<
<[ruleDescriptor.tokenListLabels,
  ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
```

```
   :{it |list_<it.label.text> = new ImplTraits::IntTrieType<CommonTokenType>(31);}; separator="\n"
>
>>

lexerRuleLabelFree() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{it |<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{it | delete list_<it.label.text>;}; separator="\n"
>
>>

ruleReturnValue() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
%>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( this->get_backtracking() > 0 ) { this->memoize(<ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

ruleCleanUp() ::= <<

// This is where rules clean up and exit
//
goto
 rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex: ;
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
<if(!ruleDescriptor.isSynPred)>
retval.call_stop_placeholder();<\n>
```

```
<endif>
<endif>
<endif>
>>

scopeClean() ::= <<
<ruleDescriptor.useScopes:{it | m_<it>_stack.pop(); }; separator="\n">
<ruleDescriptor.ruleScope:{it | m_<it.name>_stack.pop(); }; separator="\n">

>>
/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules, which do not produce tokens.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
//   Comes from: <block.description>
/** \brief Lexer rule generated by ANTLR3
*
* $ANTLR start <ruleName>
*
* Looks to match the characters the constitute the token <ruleName>
* from the attached input stream.
*
*
* \remark
*  - lexer->error == true if an exception was thrown.
*/
void <name>::m<ruleName>(<ruleDescriptor.parameterScope:parameterScope()>)
{
   ANTLR_UINT32 _type;
    <ruleDeclarations()>
   <ruleDescriptor.actions.declarations>
   <lexerRuleLabelDefs()>
   <if(trace)>
   std::cout \<\< "enter <ruleName> '" \<\< (char)this->LA(1)
             \<\< "' line=" \<\< this->getLine() \<\< ":" \<\< this->getCharPositionInLine()
             \<\< " failed=" \<\< this->get_failedflag() \<\< " backtracking=" \<\< this->get_backtracking() \<\<
std::endl;
   <endif>

<if(nakedBlock)>
   <ruleMemoization(rname=ruleName)>
   <lexerRuleLabelInit()>
   <ruleDescriptor.actions.init>

   <block><\n>
<else>
   <ruleMemoization(rname=ruleName)>
   <lexerRuleLabelInit()>
   _type     = <ruleName>;

   <ruleDescriptor.actions.init>

   <block>
 this->get_lexstate()->set_type(_type);
<endif>
   <if(trace)>
   std::cout \<\< "exit <ruleName> '" \<\< (char)this->LA(1)
            \<\< "' line=" \<\< this->getLine() \<\< ":" \<\< this->getCharPositionInLine()
            \<\< " failed=" \<\< this->get_failedflag() \<\< " backtracking=" \<\< this->get_backtracking()
 \<\< std::endl;
   <endif>
   <ruleCleanUp()>
   <lexerRuleLabelFree()>
   <(ruleDescriptor.actions.after):execAfter()>
   <memoize>
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
/** This is the entry point in to the lexer from an object that
*  wants to generate the next token, such as a pCOMMON_TOKEN_STREAM
*/
void
<name>::mTokens()
{
   <block><\n>

   goto ruleTokensEx; /* Prevent compiler warnings */
ruleTokensEx: ;
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<

// <fileName>:<description>
{
   int alt<decisionNumber>=<maxAlt>;
   <decls>
   <@predecision()>
```

```
    <decision>
    <@postdecision()>
    <@prebranch()>
    switch (alt<decisionNumber>)
    {
  <alts:{a
  | <altSwitchCase(i,a)>}>
    }
    <@postbranch()>
}
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
{
    //  <fileName>:<description>

    ANTLR_UINT32 alt<decisionNumber>;

    alt<decisionNumber>=<maxAlt>;

    <decls>
    <@predecision()>
    <decision>
    <@postdecision()>
    switch (alt<decisionNumber>)
    {
  <alts:{a | <altSwitchCase(i,a)>}>
    }
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>
```

```
/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n)
 ::= <<
// <fileName>:<description>
{
   int cnt<decisionNumber>=0;
   <decls>
   <@preloop()>

   for (;;)
   {
       int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch (alt<decisionNumber>)
 {
     <alts:{a | <altSwitchCase(i,a)>}>
     default:

  if ( cnt<decisionNumber> >= 1 )
  {
      goto loop<decisionNumber>;
  }
  <ruleBacktrackFailure()>
  <earlyExitEx()>
  <@earlyExitException()>
  goto rule<ruleDescriptor.name>Ex;
 }
 cnt<decisionNumber>++;
   }
   loop<decisionNumber>: ; /* Jump to here if this rule does not match */
   <@postloop()>
}
>>

earlyExitEx() ::= <<
/* mismatchedSetEx()
*/
new ANTLR_Exception\< <name>ImplTraits, EARLY_EXIT_EXCEPTION, StreamType>( this->get_rec(), "" );
<\n>
>>
positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
```

```
 ::= <<

// <fileName>:<description>
<decls>

<@preloop()>
for (;;)
{
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
   {
 <alts:{a | <altSwitchCase(i,a)>}>
 default:
     goto loop<decisionNumber>; /* break out of the loop */
     break;
   }
}
loop<decisionNumber>: ; /* Jump out to here if this rule does not match */
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by antlr before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase(altNum,alt) ::= <<
case <altNum>:
   <@prealt()>
   <alt>
   break;<\n>
>>

/**
 An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
```

```
{
   <@declarations()>
   <@initializations()>
   <elements:element()>
   <rew>
   <@cleanup()>
}
>>

// E L E M E N T S
/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

/** Dump the elements one per line */
element(e) ::= <<
<@prematch()>
<e.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,terminalOptions) ::= <<
<if(label)><label> = <endif> this->matchToken(<token>, &FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
list_<label>.push_back(<elem>);
>>



/** match a character */
charRef(char,label)
 ::= <<
<if(label)>
<label> = this->LA(1);<\n>
<endif>
this->matchc(<char>);
<checkRuleBacktrackFailure()>
>>
```

```
/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = this->LA(1);<\n>
<endif>
this->matchRange(<a>, <b>);
<checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,terminalOptions,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= this->LA(1);<\n>
<else>
<label>=(<labelType>) this->LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
   this->consume();
   <postmatchCode>
<if(!LEXER)>
   this->set_perror_recovery(false);
<endif>
   <if(backtracking)> this->set_failedflag(false); <\n><endif>
}
else
{
   <ruleBacktrackFailure()>
   <mismatchedSetEx()>
   <@mismatchedSetException()>
<if(LEXER)>
   this->recover();
<else>
<! use following code to make it recover inline;
   this->recoverFromMismatchedSet(&FOLLOW_set_in_<ruleName><elementIndex>);
!>
<endif>
   goto rule<ruleDescriptor.name>Ex;
}<\n>
>>

mismatchedSetEx()
 ::= <<
new ANTLR_Exception\< <name>ImplTraits, MISMATCHED_SET_EXCEPTION, StreamType>( this->get_rec(),
"" );
<if(PARSER)>
```

```
this->get_exception()->set_expectingSet(NULL);
<! use following code to make it recover inline;
this->get_exception()->set_expectingSet( &FOLLOW_set_in_<ruleName><elementIndex> );
!>
<endif>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label,elementIndex) ::= <<
<if(label)>
ANTLR_MARKER <label>Start = this->getCharIndex();
ANTLR_UINT32 <label>StartLine<elementIndex> = this->getLine();
ANTLR_UINT32 <label>StartCharPos<elementIndex> = this->getCharPositionInLine();
this->matchs(<string>);
<checkRuleBacktrackFailure()>
<label> = new CommonTokenType;
<label>->set_type( CommonTokenType::TOKEN_INVALID );
<label>->set_startIndex( <label>Start);
<label>->set_stopIndex( this->getCharIndex()-1);
<label>->set_input(
 this->get_input() );
<label>->set_line( <label>StartLine<elementIndex> );
<label>->set_charPositionInLine( <label>StartCharPos<elementIndex> );
<else>
this->matchs(<string>);
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(token,label,elementIndex,terminalOptions) ::= <<
<if(label)>
<label>=(<labelType>)this->LT(1);<\n>
<endif>
this->matchAnyToken();
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(token,label,elementIndex,terminalOptions) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>
```

```
/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = this->LA(1);<\n>
<endif>
this->matchAny();
<checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values. The 'rule' argument was the
*  target rule name, but now is type Rule,
 whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
this->followPush(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)>m_<scope:delegateName()>-><endif><rule.name>(<if(args)><args;
separator=", "><endif>);<\n>
this->followPop();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
/* <description> */
<if(label)>
{
   ANTLR_MARKER <label>Start<elementIndex> = this->getCharIndex();
   ANTLR_UINT32 <label>StartLine<elementIndex> = this->getLine();
   ANTLR_UINT32 <label>StartCharPos<elementIndex>
 = this->getCharPositionInLine();
   <if(scope)>m_<scope:delegateName()>-><endif>m<rule.name>(<if(scope)>m_<scope:delegateName()><endif>
<if(args)>, <endif><args; separator=", ">);
   <checkRuleBacktrackFailure()>
   <label> = new CommonTokenType();
   <label>->set_type( CommonTokenType::TOKEN_INVALID);
   <label>->set_startIndex( <label>Start<elementIndex> );
   <label>->set_stopIndex( this->getCharIndex()-1 );
   <label>->set_input( this->get_input() );
   <label>->set_line( <label>StartLine<elementIndex> );
   <label>->set_charPositionInLine( <label>StartCharPos<elementIndex> );
}
<else>
<if(scope)>m_<scope:delegateName()>-><endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
{
   ANTLR_UINT32 <label>Start<elementIndex>;
    ANTLR_UINT32 <label>StartLine<elementIndex> = this->getLine();
   ANTLR_UINT32 <label>StartCharPos<elementIndex> = this->getCharPositionInLine();
   <labelType> <label>;
   <label>Start<elementIndex> = this->getCharIndex();
   this->matchc(ANTLR_CHARSTREAM_EOF);
   <checkRuleBacktrackFailure()>
   <label> = new CommonTokenType();
   <label>->set_type( CommonTokenType::TOKEN_EOF );
   <label>->set_startIndex(<label>Start<elementIndex>);
   <label>->set_stopIndex(this->getCharIndex()-1);
   <label>->set_input( this->get_input() );
   <label>->set_line( <label>StartLine<elementIndex> );
   <label>->set_charPositionInLine( <label>StartCharPos<elementIndex> );
}
<else>
   this->matchc(ANTLR_CHARSTREAM_EOF);
   <checkRuleBacktrackFailure()>
   <endif>
>>
```

```
// used for left-recursive rules
recRuleDefArg()                   ::= "int <recRuleArg()>"
recRuleArg()                      ::= "_p"
recRuleAltPredicate(ruleName,opPrec)  ::= "<recRuleArg()> \<= <opPrec>"
recRuleSetResultAction()
            ::= "root_0=$<ruleName>_primary.tree;"
recRuleSetReturnAction(src,name)      ::= "$<name>=$<src>.<name>;"

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( this->LA(1)== CommonTokenType::TOKEN_DOWN ) {
   this->matchToken(CommonTokenType::TOKEN_DOWN, NULL);
   <checkRuleBacktrackFailure()>
   <children:element()>
   this->matchToken(CommonTokenType::TOKEN_UP, NULL);
   <checkRuleBacktrackFailure()>
}
<else>
this->matchToken(CommonTokenType::TOKEN_DOWN, NULL);
<checkRuleBacktrackFailure()>
<children:element()>
this->matchToken(CommonTokenType::TOKEN_UP, NULL);
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>)
 )
{
   <ruleBacktrackFailure()>
   <newFPE(...)>
}
>>

newFPE() ::= <<
 ExceptionBaseType* ex = new ANTLR_Exception\< <name>ImplTraits, FAILED_PREDICATE_EXCEPTION,
StreamType>( this->get_rec(), "<description>" );
   ex->set_ruleName( "<ruleName>" );
   <\n>
>>
```

```
// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

{
   int LA<decisionNumber>_<stateNumber> = this->LA(<k>);
   <edges; separator="\nelse ">
   else
   {
<if(eotPredictsAlt)>
       alt<decisionNumber>=<eotPredictsAlt>;
<else>
       <ruleBacktrackFailure()>

       <newNVException()>
       goto rule<ruleDescriptor.name>Ex;

<endif>
   }
}
>>

newNVException() ::= <<
ExceptionBaseType* ex = new ANTLR_Exception\< <name>ImplTraits, NO_VIABLE_ALT_EXCEPTION,
StreamType>( this->get_rec(), "<description>" );
ex->set_decisionNum( <decisionNumber> );
ex->set_state( <stateNumber> );
<@noViableAltException()>
<\n>
>>

/** Same as a normal DFA
 state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
   int LA<decisionNumber>_<stateNumber> = this->LA(<k>);
   <edges; separator="\nelse ">
}
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/

dfaLoopbackStateDecls()::= <<
ANTLR_UINT32   LA<decisionNumber>_<stateNumber>;
>>
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
  /* dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
    */
   int LA<decisionNumber>_<stateNumber> = this->LA(<k>);
   <edges; separator="\nelse "><\n>
   <if(eotPredictsAlt)>
   <if(!edges)>
 alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
 <else>
   else
   {
 alt<decisionNumber>=<eotPredictsAlt>;
   }<\n>
   <endif>
   <endif>
}
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>)<if(predicates)> && (<predicates>)<endif>)
{
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
switch ( this->LA(<k>) )
```

```
{
<edges; separator="\n">

default:
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   <newNVException()>
   goto rule<ruleDescriptor.name>Ex;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( this->LA(<k>) )
{
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( this->LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
   alt<decisionNumber>=<eotPredictsAlt>;
   break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{it |case <it>:}; separator="\n">
 {
  <targetState>
 }
   break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name>
 attribute is inherited via the parser, lexer, ...
*/
```

```
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(this, this->get_rec(), this->get_istream(),
cdfa<decisionNumber> );
<checkRuleBacktrackFailure()>
>>

/* Dump DFA tables as static initialized arrays of shorts(16 bits)/characters(8 bits)
* which are then used to statically initialize the dfa structure, which means that there
* is no runtime initialization whatsoever, other than anything the C compiler might
* need to generate. In general the C compiler will lay out memory such that there is no
* runtime code required.
*/
cyclicDFA(dfa) ::= <<
/** Static dfa state tables for Cyclic dfa:
*    <dfa.description>
*/
static const ANTLR_INT32 dfa<dfa.decisionNumber>_eot[<dfa.numberOfStates>] =
   {
 <dfa.eot; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR_INT32 dfa<dfa.decisionNumber>_eof[<dfa.numberOfStates>] =
   {
 <dfa.eof; wrap="\n", separator=", ", null="-1">
    };
static const ANTLR_INT32 dfa<dfa.decisionNumber>_min[<dfa.numberOfStates>] =
   {
 <dfa.min; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR_INT32 dfa<dfa.decisionNumber>_max[<dfa.numberOfStates>] =
   {
 <dfa.max; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR_INT32 dfa<dfa.decisionNumber>_accept[<dfa.numberOfStates>] =
   {
 <dfa.accept; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR_INT32 dfa<dfa.decisionNumber>_special[<dfa.numberOfStates>] =
   {
 <dfa.special; wrap="\n", separator=", ", null="-1">
   };

/** Used when there is no transition table entry for a particular state */
static const ANTLR_INT32* dfa<dfa.decisionNumber>_T_empty  =   NULL;

<dfa.edgeTransitionClassMap.keys:{ table |
static const ANTLR_INT32 dfa<dfa.decisionNumber>_T<i0>[] =
   {
 <table; separator=", ", wrap="\n", null="-1">
```

```
   \};<\n>}; null = "">

/* Transition tables are a table of sub tables, with some tables
* reused for efficiency.
*/
static
 const ANTLR_INT32 * const dfa<dfa.decisionNumber>_transitions[] =
{
   <dfa.transitionEdgeTables:{xref|dfa<dfa.decisionNumber>_T<xref>}; separator=", ", wrap="\n", null="NULL">
};

<@errorMethod()>

/* Declare tracking structure for Cyclic DFA <dfa.decisionNumber>
*/
class <name>CyclicDFA<dfa.decisionNumber> : public CyclicDFA\< <name>ImplTraits, <name> >, public
<name>Tokens
{
public:
 typedef CyclicDFA\< <name>ImplTraits, <name> >  BaseType;
 typedef BaseType::ContextType CtxType;

private:
<if(dfa.specialStateSTs)>
 //to maintain C-Target compatibility, we need to make some of ctx functions look like member funcs
 CtxType* m_ctx;
<endif>

public:
 <name>CyclicDFA<dfa.decisionNumber>( ANTLR_INT32 decisionNumber
     , const ANTLR_UCHAR* description
     , const ANTLR_INT32* const eot
     , const ANTLR_INT32* const eof
     , const ANTLR_INT32* const min
     , const ANTLR_INT32* const max
     , const ANTLR_INT32* const accept
     , const ANTLR_INT32* const special
     ,
 const ANTLR_INT32* const *const transition)
     :BaseType( decisionNumber, description, eot, eof, min, max, accept,
        special, transition )
 {
 <if(dfa.specialStateSTs)>
  m_ctx = NULL;
 <endif>
 }

   <if(dfa.specialStateSTs)>
   ANTLR_UINT32 LA(ANTLR_INT32 i)
   {
      return m_ctx->LA(i);
   }

   <if(PARSER)>
   const CtxType::CommonTokenType*  LT(ANTLR_INT32 k)
   {
      return m_ctx->LT(k);
   }
   <endif>
   <if(synpreds)>
   template\<typename PredType>
   bool msynpred( PredType pred )
   {
      return m_ctx->msynpred(pred);
   }
   <endif>

 ANTLR_INT32  specialStateTransition(CtxType * ctx, RecognizerType* recognizer, IntStreamType* is,
ANTLR_INT32 s)
 {
    ANTLR_INT32    _s;

    m_ctx = ctx;
    _s     = s;
    switch  (s)
    {
    <dfa.specialStateSTs:{state |
    case <i0>:

  <state>}; separator="\n">
    }
 <if(backtracking)>
    if ( ctx->get_backtracking() > 0)
    {
   ctx->set_failedflag( true );
  return -1;

   }
 <endif>
    ExceptionBaseType* ex = new ANTLR_Exception\< <name>ImplTraits, NO_VIABLE_ALT_EXCEPTION,
StreamType>( recognizer, "<dfa.description>" );
    ex->set_decisionNum( <dfa.decisionNumber> );
    ex->set_state(_s);
    <@noViableAltException()>
    return -1;
 }
 <endif>
};

static <name>CyclicDFA<dfa.decisionNumber>  cdfa<dfa.decisionNumber>(
    <dfa.decisionNumber>,       /* Decision number of this dfa      */
    /* Which decision this represents:   */
    (const ANTLR_UCHAR*)"<dfa.description>",
    dfa<dfa.decisionNumber>_eot,     /* EOT table        */
    dfa<dfa.decisionNumber>_eof,     /* EOF table        */
    dfa<dfa.decisionNumber>_min,     /* Minimum tokens for each state    */
    dfa<dfa.decisionNumber>_max,     /* Maximum tokens for each state    */
    dfa<dfa.decisionNumber>_accept, /* Accept table        */
    dfa<dfa.decisionNumber>_special, /* Special transition states      */
    dfa<dfa.decisionNumber>_transitions /* Table of transition
 tables     */

 );


/* End of Cyclic DFA <dfa.decisionNumber>
* ---------------------
*/
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
   ANTLR_UINT32 LA<decisionNumber>_<stateNumber>;<\n>
   ANTLR_MARKER index<decisionNumber>_<stateNumber>;<\n>

   LA<decisionNumber>_<stateNumber> = ctx->LA(1);<\n>
   <if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
   index<decisionNumber>_<stateNumber> = ctx->index();<\n>
   ctx->rewindLast();<\n>
   <endif>
   s = -1;
   <edges; separator="\nelse ">
 <if(semPredState)> <! return input cursor to state before we rewound !>
 ctx->seek(index<decisionNumber>_<stateNumber>);<\n>
 <endif>
   if ( s>=0 )
   {
 return s;
   }
}
break;
```

```
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr,
 targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif> )
{
   s = <targetStateNumber>;
}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "( (<left>) && (<right>) )"

orPredicates(operands) ::= "(<operands:{o|(<o>)}; separator=\"||\">)"

notPredicate(pred) ::= "!( <evalPredicate(pred,{})> )"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "this->msynpred( antlr3::ClassForwarder\<<pred>>() )"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "this->LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt)
 ::= <%
((LA<decisionNumber>_<stateNumber> >= <lower>) && (LA<decisionNumber>_<stateNumber> \<= <upper>))
%>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "((this->LA(<k>) >=
<lower>) && (this->LA(<k>) \<= <upper>))"
```

```
setTest(ranges) ::= "<ranges; separator=\" || \">"

// A T T R I B U T E S

makeScopeSet() ::= <<
/* makeScopeSet()
*/
/** Definition of the <scope.name> scope variable tracking
*  structure. An instance of this structure is created by calling
*  <name>_<scope.name>Push().
*/
struct  <scopeStruct(sname=scope.name,...)>
{
   /* =============================================================================
    * Programmer defined variables...
    */
   <scope.attributes:{it |<it.decl>;}; separator="\n">

   /* End of programmer defined variables
    * =============================================================================
    */
};

>>

globalAttributeScopeDecl(scope)
 ::= <<
<if(scope.attributes)>
/* globalAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

ruleAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

globalAttributeScopeDef(scope) ::=
<<
/* globalAttributeScopeDef(scope)
*/
<if(scope.attributes)>
```

```
StackType\< <scopeStruct(sname=scope.name)> > <scopeStack(sname=scope.name)>;

<endif>
>>

ruleAttributeScopeDef(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDef(scope)
*/
StackType\< <scopeStruct(sname=scope.name)> >  <scopeStack(sname=scope.name,...)>;

<endif>
>>

scopeStruct(sname) ::= <<
<sname>Scope
>>

scopeStack(sname) ::= <<
m_<sname>_stack
>>

returnStructName(r) ::= "<r.name>_return"

returnType() ::= <%
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>::<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
bool
<endif>
%>

/**
 Generate the C type associated with a single or multiple return
*  value(s).
*/
ruleLabelType(referencedRule) ::= <%
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
```

```
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
%>

delegateName(d) ::= <<
<if(d.label)><d.label><else>g<d.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "0".
*/
initValue(typeName) ::= <<
= <cTypeInitMap.(typeName)>
>>

/** Define a rule label  */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text>;
>>
/**  Rule label default value */
ruleLabelInitVal(label) ::= <<
>>

ASTLabelType() ::=
"<if(recognizer.ASTLabelType)><recognizer.ASTLabelType><else>ImplTraits::TreeType*<endif>"

/** Define a return struct for a rule if the code
 needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
struct <ruleDescriptor:returnStructName()> : public <name>ImplTraits::RuleReturnValueType
{
public:
   typedef <name>ImplTraits::RuleReturnValueType BaseType;
   <ruleDescriptor:returnStructName()>()
       : BaseType()
       <if(scope)>, <scope.attributes:{it | <it.name>() }; separator=","><endif>
       { init(); }
   <ruleDescriptor:returnStructName()>( BaseParserType* parser )
```

```
        : BaseType(parser)
        <if(scope)>, <scope.attributes:{it | <it.name>() }; separator=","><endif>
        { init(); }
    <ruleDescriptor:returnStructName()>( const <ruleDescriptor:returnStructName()>& other )
        : BaseType(other)
    <if(scope)>, <scope.attributes:{it | <it.name>(other.<it.name>) }; separator=", "><endif>
    { copy(other);
  }
    ~<ruleDescriptor:returnStructName()>()
    {
        <@ruleReturnMembersDelete()>
    }

    <ruleDescriptor:returnStructName()>&
    operator=( const <ruleDescriptor:returnStructName()>& other )
    {
        BaseType::operator=( other );
        <if(scope)><scope.attributes:{it | <it.name> = other.<it.name>; }; separator="\n"><endif>
        copy(other);
     return *this;
    }
    <@ruleReturnMembers()>
    void init() { <@ruleReturnMembersInit()> }
    void copy( const <ruleDescriptor:returnStructName()>& other) { <@ruleReturnMembersCopy()> }
<else>
struct <ruleDescriptor:returnStructName()>
{
public:
    <name>ImplTraits::<recognizer.ASTLabelType>       start;
    <name>ImplTraits::<recognizer.ASTLabelType>       stop;
    <ruleDescriptor:returnStructName()>( const <ruleDescriptor:returnStructName()>& other )
    <if(scope.attributes)>
    <scope.attributes:{it | <it.name>(other.<it.name>) }; separator=",">
    <endif>
    {
      start = other.start;
      stop
 = other.stop;
    }

    <ruleDescriptor:returnStructName()>&
    operator=( const <ruleDescriptor:returnStructName()>& other )
    {
      start = other.start;
      stop  = other.stop;

     <scope.attributes:{it | <it.name> = other.<it.name>; }; separator="\n">
     return *this;
```

```
   }
<endif>
   <if(scope)><scope.attributes:{it |<it.type> <it.name>;}; separator="\n"><endif>
};

<endif>
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{it |<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name>=<expr>;"

/** Note that the scopeAttributeRef does not have access to the
* grammar name directly
*/
scopeAttributeRef(scope,attr,index,negIndex) ::= <%
<if(negIndex)>
 m_<scope>_stack.at( m_<scope>_stack.size()-<negIndex>-1).<attr.name>
<else>
<if(index)>
 m_<scope>_stack.at(<index>).<attr.name>
<else>
 m_<scope>_stack.peek().<attr.name>
<endif>
<endif>
%>

scopeSetAttributeRef(scope,attr,expr,index,negIndex)
 ::= <%
<if(negIndex)>
 m_<scope>_stack.at( m_<scope>_stack.size()-<negIndex>-1).<attr.name> = <expr>;
<else>
<if(index)>
 m_<scope>_stack.at(<index>).<attr.name> = <expr>;
<else>
 m_<scope>_stack.peek().<attr.name> =<expr>;
<endif>
<endif>
%>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
```

```
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<scope>.<attr.name>
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>=<expr>;
<else>
<attr.name>=<expr>;
<endif>
>>

/**
 How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach
//
tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>->getText())"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>->get_type())"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>->get_line())"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>->get_charPositionInLine())"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>->get_channel())"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>->get_tokenIndex())"
tokenLabelPropertyRef_tree(scope,attr) ::= "(<scope>->get_tree())"
tokenLabelPropertyRef_int(scope,attr) ::= "(<name>ImplTraits::ConvertToInt32(<scope>->getText()))"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>.start)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>.stop)"
```

```
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>.tree)"
ruleLabelPropertyRef_text(scope,attr)
 ::= <<
<if(TREE_PARSER)>
(this->get_strstream()->toStringSS(<scope>.start, <scope>.start))
<else>
(this->get_strstream()->toStringTT(<scope>.start, <scope>.stop))
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(<scope>->get_type())"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(<scope>->get_line())"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(<scope>->get_charPositionInLine())"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(<scope>->get_channel())"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(<scope>->get_tokenIndex())"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(<scope>->getText())"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop"
rulePropertyRef_tree(scope,attr)
 ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
this->get_input()->toStringSS( this->get_adaptor()->getTokenStartIndex(retval.start), this->get_adaptor()->getTokenStopIndex(retval.start))
<else>
this->get_strstream()->toStringTT(retval.start, this->LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "this->getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "this->get_state()->get_tokenStartLine()"
lexerRulePropertyRef_pos(scope,attr) ::= "this->get_state()->get_tokenStartCharPositionInLine()"
lexerRulePropertyRef_channel(scope,attr) ::= "this->get_state()->get_channel()"
lexerRulePropertyRef_start(scope,attr) ::= "this->get_state()->get_tokenStartCharIndex()"
lexerRulePropertyRef_stop(scope,attr) ::= "(this->getCharIndex()-1)"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_int(scope,attr)
 ::= "(<name>ImplTraits::ConvertToInt32(<scope>->getText()))"
```

```
// setting $st and $tree is allowed in local rule. everything else is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree=<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st=<expr>;"


/** How to deal with an @after for C targets. Because we cannot rely on
*  any garbage collection, after code is executed even in backtracking
*  mode. Must be documented clearly.
*/
execAfter(action) ::= <<
{
   <action>
}
>>

/** How to execute an action (when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> )
{
   <action>
}
<else>
if ( BACKTRACKING == 0 )
{
   <action>
}
<endif>
<else>
{
   <action>
}
<endif>
>>

// M I S C (properties, etc...)

bitsetDeclare(bitsetname, words64, traits) ::= <<

/** Bitset defining follow set for error recovery in rule state: <name>  */
static ANTLR_BITWORD
 <bitsetname>_bits[] = { <words64:{it |ANTLR_UINT64_LIT(<it>)}; separator=", "> };
static  <traits>::BitsetListType <bitsetname>( <bitsetname>_bits, <length(words64)> );
>>

codeFileExtension() ::= ".cpp"
```

true_value() ::= "true"
false_value() ::= "false"

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/Cpp/Cpp.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAConversionThread.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/RuleLabelScope.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/State.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Interp.java
*
 /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/Transition.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/LL1Analyzer.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-

jar/org/antlr/tool/GrammarSyntaxMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/PredicateLabel.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/Tool.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAContext.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/Barrier.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NonLLStarDecisionException.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/RuleClosureTransition.java
*
 /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Message.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/PythonTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Rule.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/Utils.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/BitSet.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarReport.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DFAOptimizer.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/JavaScriptTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAState.java
*
 /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/RecursionOverflowMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSemanticsMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/AttributeScope.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/ToolMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/ActionScriptTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSerializerFoo.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarUnreachableAltsMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/DOTGenerator.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/AssignTokenTypesBehavior.java
*
 /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAConfiguration.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarAnalysisAbortedMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/LL1DFA.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/RandomPhrase.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/OrderedHashSet.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/BuildDependencyGenerator.java

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarInsufficientPredicatesMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/MutableInteger.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/NonRegularDecisionMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/Perl5Target.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/ANTLRErrorListener.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/NameSpaceChecker.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/StateCluster.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarNonDeterminismMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Strip.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/CompositeGrammarTree.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DFAState.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Interpreter.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/Target.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/IntArrayList.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/FASerializer.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFAToDFAConverter.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/Interval.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CSharp2Target.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/ACyclicDFACodeGenerator.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/MultiMap.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/JavaTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/DelphiTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSpelunker.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/AnalysisTimeoutException.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/LeftRecursionCyclesMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarDanglingStateMessage.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/Label.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DecisionProbe.java
*

/opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/DFA.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/MachineProbe.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/SemanticContext.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/Python3Target.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/CompositeGrammar.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/IntSet.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/AnalysisRecursionOverflowException.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarReport2.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/IntervalSet.java
*
/opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarAST.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/NFAFactory.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/NFA.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/ActionLabel.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Grammar.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/GrammarSanity.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/analysis/LookaheadSet.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/CppTarget.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/Attribute.java
* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/tool/ErrorManager.java
*
/opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/misc/Graph.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
* Copyright (c) 2007-2008 Johannes Luber
* Copyright (c) 2005-2007 Kunle Odutola
* Copyright (c) 2011 Sam Harwell
* Copyright (c) 2011 Terence Parr
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
```

```
/** Template overrides to add debugging to normal Java output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/

@outputFile.debugPreprocessor() ::= "#define ANTLR_DEBUG"

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.Runtime.Debug;
using IOException = System.IO.IOException;
>>

@genericParser.members()
 ::= <<
<if(grammar.grammarIsRoot)>
public static readonly string[] ruleNames =
 new string[]
 {
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n ", separator=", ">
 };<\n>
<endif>
<if(grammar.grammarIsRoot)><! grammar imports other grammar(s) !>
int ruleLevel = 0;
public virtual int RuleLevel { get { return ruleLevel; } }
public virtual void IncRuleLevel() { ruleLevel++; }
public virtual void DecRuleLevel() { ruleLevel--; }
<if(profile)>
<ctorForProfilingRootGrammar()>
<else>
<ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else><! imported grammar !>
public int RuleLevel { get { return <grammar.delegators:{g| <g:delegateName()>}>.RuleLevel; } }
public void IncRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.IncRuleLevel(); }
public void DecRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.DecRuleLevel(); }
<ctorForDelegateGrammar()>
<endif>
<if(profile)>
public override bool AlreadyParsedRule( IIntStream input, int
 ruleIndex )
{
 int stopIndex = GetRuleMemoization(ruleIndex, input.Index);
 ((Profiler)dbg).ExamineRuleMemoization(input, ruleIndex, stopIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
 return base.AlreadyParsedRule(input, ruleIndex);
}<\n>
public override void Memoize( IIntStream input, int ruleIndex, int ruleStartIndex )
{
 ((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
 base.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected virtual bool EvalPredicate( bool result, string predicate )
{
 dbg.SemanticPredicate( result, predicate );
 return result;
}<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
 : this( input, DebugEventSocketProxy.DefaultDebuggerPort, new
 RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, int port,
RecognizerSharedState state )
 : base( input, state )
{
 <createListenerAndHandshake()>
 <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}> );}; separator="\n">
 <parserCtorBody()>
 <@finally()>
}<\n>
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
```

```
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input )
 : this( input, new Profiler(null), new RecognizerSharedState() )
{
}
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state )
 : base( input, dbg, state )
{
 Profiler p = (Profiler)dbg;
 p.setParser(this);
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state,
 this<grammar.delegators:{g|, <g:delegateName()>}> );}; separator="\n">
 <parserCtorBody()>
 <@finally()>
}
<\n>
>>

/** Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg,
RecognizerSharedState state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}> )
 : base( input, dbg, state )
{
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}> );}; separator="\n">
 <parserCtorBody()>
}<\n>
>>

ctorForPredefinedListener() ::= <<
<actions.(actionScope).ctorModifier; null="public"> <name>( <inputStreamType> input, IDebugEventListener dbg
)
 <@superClassRef>: base( input, dbg, new RecognizerSharedState() )<@end>
{
<if(profile)>
 Profiler p = (Profiler)dbg;
 p.setParser(this);
<endif>
 <grammar.directDelegates:{g|<g:delegateName()>
 = new <g.recognizerName>(input, dbg, this.state, this<grammar.delegators:{g|, <g:delegateName()>}>);};
separator="\n">
 <parserCtorBody()>
 <@finally()>
}<\n>
>>
```

```
createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, input.TreeAdaptor );<\n>
<else>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, null );<\n>
<endif>
DebugListener = proxy;
try
{
 proxy.Handshake();
}
catch ( IOException ioe )
{
 ReportError( ioe );
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

/*
* Many of the following rules were merged into CSharp2.stg.
*/

@rule.preamble() ::= <<
if (RuleLevel == 0)
 DebugListener.Commence();
IncRuleLevel();
>>
//@rule.preamble() ::= <<
//try
//{
// dbg.EnterRule( GrammarFileName, "<ruleName>" );
// if ( RuleLevel == 0 )
// {
//  dbg.Commence();
// }
// IncRuleLevel();
// dbg.Location( <ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine> );<\n>
//>>

@rule.postamble()
 ::= <<
DecRuleLevel();
if (RuleLevel == 0)
 DebugListener.Terminate();
>>
//@rule.postamble() ::= <<
//dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
//}
//finally
//{
// dbg.ExitRule( GrammarFileName, "<ruleName>" );
// DecRuleLevel();
// if ( RuleLevel == 0 )
// {
//  dbg.Terminate();
// }
//}<\n>
//>>

//@insertSynpreds.start() ::= "dbg.BeginBacktrack( state.backtracking );"
//@insertSynpreds.stop() ::= "dbg.EndBacktrack( state.backtracking, success );"

// Common debug event triggers used by region overrides below

//enterSubRule() ::= <<
//try
//{
// dbg.EnterSubRule( <decisionNumber> );<\n>
//>>

//exitSubRule() ::= <<
//}
//finally
//{
// dbg.ExitSubRule( <decisionNumber> );
//}<\n>
//>>

//enterDecision() ::= <<
//try
//{
// dbg.EnterDecision( <decisionNumber> );<\n>
//>>

//exitDecision() ::= <<
//}
//finally
//{
// dbg.ExitDecision( <decisionNumber> );
//}<\n>
//>>

//enterAlt(n) ::= "dbg.EnterAlt( <n> );<\n>"
```

```
//
 Region overrides that tell various constructs to add debugging triggers

//@block.predecision() ::= "<enterSubRule()><enterDecision()>"

//@block.postdecision() ::= "<exitDecision()>"

//@block.postbranch() ::= "<exitSubRule()>"

//@ruleBlock.predecision() ::= "<enterDecision()>"

//@ruleBlock.postdecision() ::= "<exitDecision()>"

//@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

//@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

//@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

//@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

//@positiveClosureBlock.predecision() ::= "<enterDecision()>"

//@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

//@positiveClosureBlock.earlyExitException() ::=
// "dbg.RecognitionException( eee<decisionNumber> );<\n>"

//@closureBlock.preloop() ::= "<enterSubRule()>"

//@closureBlock.postloop() ::= "<exitSubRule()>"

//@closureBlock.predecision() ::= "<enterDecision()>"

//@closureBlock.postdecision() ::= "<exitDecision()>"

//@altSwitchCase.prealt()
 ::= "<enterAlt(n=i)>"

//@element.prematch() ::=
// "dbg.Location( <it.line>, <it.pos> );"

//@matchSet.mismatchedSetException() ::=
// "dbg.RecognitionException( mse );"

//@dfaState.noViableAltException() ::= "dbg.RecognitionException( nvae );"

//@dfaStateSwitch.noViableAltException() ::= "dbg.RecognitionException( nvae );"
```

```
//dfaDecision(decisionNumber,description) ::= <<
//try
//{
// isCyclicDecision = true;
// <super.dfaDecision(...)>
//}
//catch ( NoViableAltException nvae )
//{
// dbg.RecognitionException( nvae );
// throw nvae;
//}
//>>

//@cyclicDFA.errorMethod() ::= <<
//public override void Error( NoViableAltException nvae )
//{
// ((DebugParser)recognizer).dbg.RecognitionException( nvae );
//}
//>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
EvalPredicate(<pred>, "<description>")
>>
```

Found in path(s):

* /opt/cola/permits/172853067_1695318043.7771866/0/antlr-3-5-2-sources-jar/org/antlr/codegen/templates/CSharp2/Dbg.stg

1.50 backport-util-concurrent 3.1

1.50.1 Available under license :

The backport-util-concurrent software has been released to the public domain,
as explained at: http://creativecommons.org/licenses/publicdomain.

Acknowledgements: backport-util-concurrent is based in large part on the public
domain sources from:
1) JSR166,
2) package dl.util.concurrent,
3) Doug Lea's "collections" package.

1.51 commons-digester 1.8.1

1.51.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache Commons Digester
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.52 commons-io 1.4

1.52.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

Apache Commons IO
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.53 hibernate 3.6.3

1.54 javax-websocket-api 1.1

1.54.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL

Found in path(s):
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/DecodeException.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/PathParam.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/ServerContainer.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/SendResult.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/DefaultClientEndpointConfig.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/Encoder.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/Decoder.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/RemoteEndpoint.java
*
 /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/ClientEndpoint.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/package-info.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/HandshakeRequest.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/CloseReason.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/ServerEndpointConfig.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/DefaultServerEndpointConfig.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/WebSocketContainer.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/ClientEndpointConfig.java
*
 /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/Endpoint.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-

jar/javax/websocket/server/ServerApplicationConfig.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/OnOpen.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/package-info.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/EndpointConfig.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/DeploymentException.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/ContainerProvider.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/MessageHandler.java
*
 /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/OnError.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/SendHandler.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/OnClose.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/HandshakeResponse.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/EncodeException.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/server/ServerEndpoint.java
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/OnMessage.java
No license file was found, but licenses were detected in source scan.

* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/Session.java
No license file was found, but licenses were detected in source scan.

<!--
 DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

 Copyright (c) 2012-2014 Oracle and/or its affiliates. All rights reserved.

 The contents of this file are subject to the terms of either the GNU
 General Public License Version 2 only ("GPL") or the Common Development
 and Distribution License("CDDL") (collectively, the "License"). You
 may not use this file except in compliance with the License. You can
 obtain a copy of the License at
 http://glassfish.java.net/public/CDDL+GPL_1_1.html
 or packager/legal/LICENSE.txt. See the License for the specific
 language governing permissions and limitations under the License.

 When distributing the software, include this License Header Notice in each
 file and include the License file at packager/legal/LICENSE.txt.

 GPL Classpath Exception:
 Oracle designates this particular file as subject to the "Classpath"

exception as provided by Oracle
in the GPL Version 2 section of the License
file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields
enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license." If you don't indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above. However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if the new code is made subject to such option by the copyright
holder.
-->

Found
in path(s):
* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/META-INF/maven/javax.websocket/javax.websocket-api/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception as provided

```
 by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to
 such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/172872113_1695318074.4608743/0/javax-websocket-api-1-1-sources-jar/javax/websocket/PongMessage.java

1.55 joda-time 2.2

1.55.1 Available under license :

```
=============================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================
```

This product includes software developed by
Joda.org (http://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.56 junit 4.10

1.56.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Copyright 2010 Google Inc. All Rights Reserved.

Found in path(s):
* /opt/cola/permits/172881795_1695318087.7769952/0/junit-4-10-sources-jar/org/junit/runners/model/MultipleFailureException.java

1.57 apache-log4j 2.8.2

1.57.1 Available under license :

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

limitations under the License.

1.58 logback-core 1.1.7

1.59 metrics-core 2.2.0

1.59.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/main/java/util/concurrent/ThreadLocalRandom.java?view=markup
*/
```

Found in path(s):

* /opt/cola/permits/172884834_1695318090.763587/0/metrics-core-2-2-0-sources-jar/com/yammer/metrics/stats/ThreadLocalRandom.java

1.60 spring-aop 3.0.5

1.60.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2087 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DelegatePerTargetObjectIntroductionInterceptor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/InstantiationModelAwarePointcutAdvisorImpl.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/AbstractBeanFactoryBasedTargetSource.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/AbstractLazyCreationTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/ThreadLocalTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/ProxyConfig.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/RuntimeTestWalker.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/HotSwappableTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AdvisedSupport.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/BeanNameAutoProxyCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AopConfigUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/SingletonTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AopProxyUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-

jar/org/springframework/aop/framework/AopContext.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/ExposeBeanNameAdvisors.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/dynamic/AbstractRefreshableTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AopNamespaceUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJExpressionPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/AbstractAspectJAdvisorFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/EmptyTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJAfterReturningAdvice.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/Cglib2AopProxy.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/AbstractBeanFactoryPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AbstractInterceptorDrivenBeanDefinitionDecorator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/AopUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AbstractAspectJAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/scope/ScopedProxyUtils.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/AutoProxyUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/JdkDynamicAopProxy.java
No license file was found, but licenses were detected in source scan.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/TrueMethodMatcher.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/ControlFlowPointcut.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/TrueClassFilter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/target/QuickTargetSourceCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/AbstractExpressionPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/PointcutEntry.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AdvisorEntry.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/InstantiationModelAwarePointcutAdvisor.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AdviceEntry.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJExpressionPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/TruePointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/NameMatchMethodPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJPrecedenceInformation.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/target/LazyInitTargetSourceCreator.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/dynamic/BeanFactoryRefreshableTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/PerformanceMonitorInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/ExpressionPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AopConfigException.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJMethodBeforeAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/AopInvocationException.java

* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/TypePatternClassFilter.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/JamonPerformanceMonitorInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/DefaultAdvisorAutoProxyCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/DebugInterceptor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DynamicMethodMatcher.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/ThreadLocalTargetSourceStats.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/RootClassFilter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/PointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/ConcurrencyThrottleInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/UnknownAdviceTypeException.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/StaticMethodMatcher.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/PoolingConfig.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/GlobalAdvisorAdapterRegistry.java

* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DynamicMethodMatcherPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/MethodBeforeAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/TargetSourceCreator.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/ConfigBeanDefinitionParser.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/BeanFactoryAdvisorRetrievalHelper.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/DefaultAdvisorChainFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AdvisorComponentDefinition.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/AbstractTraceInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/StaticMethodMatcherPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/autoproxy/AspectJAwareAdvisorAutoProxyCreator.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DefaultIntroductionAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/DefaultAdvisorAdapterRegistry.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-

jar/org/springframework/aop/framework/ProxyCreatorSupport.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/aspectj/annotation/BeanFactoryAspectJAdvisorsBuilder.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/ThrowsAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/framework/adapter/ThrowsAdviceInterceptor.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/support/NameMatchMethodPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/aspectj/AspectJAdviceParameterNameDiscoverer.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/framework/autoproxy/target/AbstractBeanFactoryBasedTargetSourceCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/aspectj/AspectJProxyUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/interceptor/CustomizableTraceInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/framework/AdvisorChainFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2007 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/aspectj/annotation/ReflectiveAspectJAdvisorFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/IntroductionAdvisor.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/config/SimpleBeanFactoryAwareAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-
jar/org/springframework/aop/config/SpringConfiguredBeanDefinitionParser.java

* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/TargetClassAware.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/DeclareParentsAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/StaticMethodMatcherPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/Pointcuts.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AopProxy.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AspectComponentDefinition.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/Advisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/IntroductionAwareMethodMatcher.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AopProxyFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/LazyInitTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AdvisedSupportListener.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/Pointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/AdvisorAdapter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJAfterAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJAfterThrowingAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DefaultPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/SimpleBeanTargetSource.java
*
 /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/IntroductionInfo.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/InterceptorAndDynamicMethodMatcher.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/AdvisorAdapterRegistrationManager.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/SimpleTraceInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/RawTargetAccess.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-

jar/org/springframework/aop/aspectj/annotation/MetadataAwareAspectInstanceFactory.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/AfterReturningAdviceAdapter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/ComposablePointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/DefaultAopProxyFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/AbstractPoolingTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/ProxyMethodInvocation.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/AfterAdvice.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/DynamicIntroductionAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/AdvisorAdapterRegistry.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AopNamespaceHandler.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/AfterReturningAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DefaultBeanFactoryPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/DelegatingIntroductionInterceptor.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/annotation/AnnotationClassFilter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/NotAnAtAspectException.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/PrototypeAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/MethodBeforeAdviceAdapter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/PrototypeTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/annotation/AnnotationMethodMatcher.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/MethodMatchers.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AspectEntry.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/annotation/AnnotationMatchingPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-

jar/org/springframework/aop/support/ClassFilters.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJAopUtils.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/AbstractGenericPointcutAdvisor.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJWeaverMessageHandler.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/SpringProxy.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/scope/ScopedObject.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/PointcutComponentDefinition.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/ClassFilter.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/AspectJAdvisorFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/InfrastructureAdvisorAutoProxyCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/AspectJAroundAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/MethodMatcher.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/Advised.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/RegexpMethodPointcutAdvisor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/IntroductionInterceptor.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/ThrowsAdviceAdapter.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AopInfrastructureBean.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/AbstractMonitoringInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/BeforeAdvice.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/AfterReturningAdviceInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/CommonsPoolTargetSource.java
*

/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/adapter/MethodBeforeAdviceInterceptor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/AspectJProxyFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/AbstractPointcutAdvisor.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/AbstractPrototypeBasedTargetSource.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/AspectJAutoProxyBeanDefinitionParser.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/scope/ScopedProxyFactoryBean.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/autoproxy/AspectJPrecedenceComparator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/ProxyFactoryBean.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/IntroductionInfoSupport.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/AbstractAutoProxyCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/SimpleAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/ExposeInvocationInterceptor.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/AbstractRegexpMethodPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-

jar/org/springframework/aop/aspectj/annotation/BeanFactoryAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/MethodLocatingFactoryBean.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/LazySingletonAspectInstanceFactoryDecorator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/ReflectiveMethodInvocation.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/ProxyFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/ProxyCreationContext.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/autoproxy/AbstractAdvisorAutoProxyCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/MethodInvocationProceedingJoinPoint.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/support/JdkRegexpMethodPointcut.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/SingletonAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/AspectMetadata.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/framework/AbstractSingletonProxyFactoryBean.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/scope/DefaultScopedObject.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/config/ScopedProxyBeanDefinitionDecorator.java
*
/opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/SimpleMetadataAwareAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/SingletonMetadataAwareAspectInstanceFactory.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/aspectj/annotation/AnnotationAwareAspectJAutoProxyCreator.java
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/interceptor/AsyncExecutionInterceptor.java
No license file was found, but licenses were detected in source scan.

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/target/dynamic/Refreshable.java
No license file was found, but licenses were detected in source scan.

/*<
* Copyright 2002-2010 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/172893870_1695318093.4852884/0/org-springframework-aop-sources-3-0-5-release-2-jar/org/springframework/aop/TargetSource.java

1.61 spring-beans 3.0.5

1.61.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2006 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeanInstantiationException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanClassLoaderAware.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/AbstractComponentDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/Location.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanReference.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanInitializationException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/ByteArrayPropertyEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanCurrentlyInCreationException.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/BeanDefinitionDocumentReader.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ComponentDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/SourceExtractor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/XmlBeanDefinitionStoreException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/MethodInvocationException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/PropertyEntry.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/URLEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/HierarchicalBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/BeanEntry.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/BeanDefinitionParsingException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-

jar/org/springframework/beans/factory/xml/BeanDefinitionParser.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/PassThroughSourceExtractor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/NamedBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/FatalBeanException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/NamespaceHandlerResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/InvalidPropertyException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/NullSourceExtractor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/BeanDefinitionDecorator.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/Mergeable.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/NotReadablePropertyException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/ResourceBundleEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/NullValueInNestedPathException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanCreationNotAllowedException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CharArrayPropertyEditor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeanMetadataElement.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/CommonsLogFactoryBean.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2010 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/XmlBeanDefinitionReader.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/support/ResourceEditorRegistrar.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ObjectFactoryCreatingFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/UUIDEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/support/PagedListHolder.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/MutablePropertyValues.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeanUtils.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/SimpleAutowireCandidateResolver.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/InputStreamEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ConstructorArgumentValues.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/PropertiesFactoryBean.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AbstractBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ProviderCreatingFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/DefaultSingletonBeanRegistry.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/DirectFieldAccessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/DeprecatedBeanWarner.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/TypeConverterDelegate.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanIsAbstractException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/TypedStringValue.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanDefinitionVisitor.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/DisposableBeanAdapter.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanFactoryUtils.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CharsetEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/FactoryBeanRegistrySupport.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/MethodOverrides.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/InputSourceEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/SimpleInstantiationStrategy.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeanWrapperImpl.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionValueResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/InjectionMetadata.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/InstantiationStrategy.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/BeanDefinitionParserDelegate.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/CachedIntrospectionResults.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/ClassEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyEditorRegistrySupport.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/SimpleTypeConverter.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionRegistryPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/URIEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/PropertyPlaceholderConfigurer.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AutowireUtils.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/DefaultBeanDefinitionDocumentReader.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/DefaultListableBeanFactory.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AbstractBeanDefinitionReader.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/RootBeanDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AbstractAutowireCapableBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionReader.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/FactoryBean.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/PropertyResourceConfigurer.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/AutowiredAnnotationBeanPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/ClassArrayEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ConstructorResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyAccessorUtils.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/AbstractFactoryBean.java
*

/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/PropertyPathFactoryBean.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/XmlBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CustomBooleanEditor.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/RuntimeBeanNameReference.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/serviceloader/AbstractServiceLoaderBasedFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanDefinitionStoreException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ReaderEventListener.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/serviceloader/ServiceFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionDefaults.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/AnnotationBeanWiringInfoResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanReferenceFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanFactoryAware.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/CglibSubclassingInstantiationStrategy.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-

jar/org/springframework/beans/factory/xml/NamespaceHandler.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ProblemReporter.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ReaderContext.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ManagedProperties.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/CannotLoadBeanClassException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/ResourceEntityResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/PreferencesPlaceholderConfigurer.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/StringTrimmerEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/access/BeanFactoryReference.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ImportDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/LookupOverride.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/DefaultsDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/AnnotatedBeanDefinition.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/serviceloader/ServiceLoaderFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanNameAware.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyEditorRegistry.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AutowireCandidateQualifier.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/AbstractSimpleBeanDefinitionParser.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyValuesEditor.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ConstructorArgumentEntry.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/DefaultDocumentLoader.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyEditorRegistrar.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-

jar/org/springframework/beans/BeanMetadataAttribute.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/QualifierEntry.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeanMetadataAttributeAccessor.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/Required.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/FactoryBeanNotInitializedException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/access/BeanFactoryLocator.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/AliasDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/wiring/BeanWiringInfo.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/FailFastProblemReporter.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/XmlReaderContext.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/wiring/ClassNameBeanWiringInfoResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanNotOfRequiredTypeException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanNameGenerator.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/RuntimeBeanReference.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/Configurable.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/wiring/BeanWiringInfoResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/UtilNamespaceHandler.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionValidationException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/MethodOverride.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/PatternEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/DefaultBeanNameGenerator.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/support/MutableSortDefinition.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/MethodReplacer.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanCreationException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/serviceloader/ServiceListFactoryBean.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanExpressionException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-

jar/org/springframework/beans/factory/support/BeanDefinitionResource.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/ObjectFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyMatches.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/NotWritablePropertyException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/GenericBeanDefinition.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CustomDateEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/SmartFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeansException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ListFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/ParseState.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/MergedBeanDefinitionPostProcessor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/Problem.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/UnsatisfiedDependencyException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionRegistry.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/StringArrayPropertyEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/Scope.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ChildBeanDefinition.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/support/PropertyComparator.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/access/SingletonBeanFactoryLocator.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/access/el/SimpleSpringBeanELResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/AnnotatedGenericBeanDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/SmartInstantiationAwareBeanPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/PropertiesBeanDefinitionReader.java

*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/support/ArgumentConvertingMethodInvoker.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/InstantiationAwareBeanPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/CompositeComponentDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyAccessorFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/MapFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/BeanComponentDefinition.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/Autowired.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/access/el/SpringBeanELResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ConfigurableListableBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/SimpleBeanDefinitionRegistry.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CharacterEditor.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/SetFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanExpressionResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/DestructionAwareBeanPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/Qualifier.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/CustomAutowireConfigurer.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/annotation/AnnotationBeanUtils.java
*
/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CustomMapEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/AbstractSingleBeanDefinitionParser.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ReplaceOverride.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionReaderUtils.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/GenericTypeAwarePropertyDescriptor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/StaticListableBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/Autowire.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/FieldRetrievingFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/BeansDtdResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/NamespaceHandlerSupport.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/ConfigurablePropertyAccessor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AbstractBeanDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/DocumentDefaultsDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/TimeZoneEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ConfigurableBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/DefaultNamespaceHandlerResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/AbstractPropertyAccessor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanDefinitionHolder.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyValue.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/CustomScopeConfigurer.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ManagedSet.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/MethodInvokingFactoryBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/DependencyDescriptor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyBatchUpdateException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/ListableBeanFactory.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/parsing/EmptyReaderEventListener.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/PluggableSchemaResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanExpressionContext.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CurrencyEditor.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CustomCollectionEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/AutowireCandidateResolver.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/BeanWrapper.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/ConversionNotSupportedException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/TypeMismatchException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/DelegatingEntityResolver.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/CustomNumberEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/PropertyOverrideConfigurer.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/FileEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/ParserContext.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-

jar/org/springframework/beans/factory/config/SingletonBeanRegistry.java

*

/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/PropertiesEditor.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/wiring/BeanConfigurerSupport.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ManagedMap.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/InstantiationAwareBeanPostProcessorAdapter.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/RequiredAnnotationBeanPostProcessor.java

*

/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyAccessException.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ManagedArray.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyAccessor.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/PropertyValues.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/NoSuchBeanDefinitionException.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/QualifierAnnotationAutowireCandidateResolver.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/AutowireCapableBeanFactory.java

*

/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/Value.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/AbstractBeanDefinitionParser.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/DocumentLoader.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/xml/SimplePropertyNamespaceHandler.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/ServiceLocatorFactoryBean.java

*

/opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/annotation/InitDestroyAnnotationBeanPostProcessor.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/ManagedList.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/TypeConverter.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/BeanDefinitionBuilder.java

* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-

jar/org/springframework/beans/factory/config/CustomEditorConfigurer.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/support/SortDefinition.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/BeanIsNotAFactoryException.java
*
 /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/access/BootstrapException.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/config/BeanFactoryPostProcessor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/propertyeditors/LocaleEditor.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/InitializingBean.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/DisposableBean.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/SimpleSecurityContextProvider.java
* /opt/cola/permits/172894262_1695318096.375178/0/org-springframework-beans-sources-3-0-5-release-2-jar/org/springframework/beans/factory/support/SecurityContextProvider.java

1.62 spring-context 3.0.5

1.62.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/access/DefaultLocatorFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiTemplateEditor.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/NoSuchMessageException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/HierarchicalMessageSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/HierarchicalThemeSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/MessageSourceAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/context/access/ContextBeanFactoryReference.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/weblogic/WebLogicClassLoaderAdapter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/weblogic/WebLogicLoadTimeWeaver.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/jboss/JBossTranslatorAdapter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/oc4j/OC4JClassPreprocessorAdapter.java
*

/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/jboss/JBossClassLoaderAdapter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/oc4j/OC4JClassLoaderAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/AbstractApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/TaskExecutorFactoryBean.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/support/FormattingConversionServiceFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/DefaultBindingErrorProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/JodaDateTimeFormatAnnotationFormatterFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/MessageSourceResourceBundleLocator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/MessageSourceResolvable.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/AutodetectCapableMBeanInfoAssembler.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiLocatorDelegate.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/DataBinder.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiObjectLocator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/context/annotation/ConfigurationClassPostProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/LocalValidatorFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/TaskScheduler.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ConfigurationClassBeanDefinitionReader.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/AbstractMessageSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/CronTrigger.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ConversionServiceFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/MessageSourceSupport.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/AbstractMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/AbstractReflectiveMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ConfigurationClass.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/expression/StandardBeanExpressionResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/FileSystemXmlApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/AnnotationDrivenBeanDefinitionParser.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/InstrumentationLoadTimeWeaver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/AbstractApplicationEventMulticaster.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/SpringValidatorAdapter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/annotation/ScheduledAnnotationBeanPostProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/EmbeddedValueResolverAware.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/weaving/DefaultContextLoadTimeWeaver.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/number/PercentFormatter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ClassPathXmlApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/jndi/JndiLocatorSupport.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/annotation/Scheduled.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/CustomValidatorBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/JodaTimeContextHolder.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/MBeanClientInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ConfigurationClassEnhancer.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/number/NumberFormatAnnotationFormatterFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/AbstractPropertyBindingResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/i18n/LocaleContextHolder.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/expression/BeanFactoryResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/MetadataMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/number/CurrencyFormatter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/expression/MapAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/AbstractBindingResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiObjectFactoryBean.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/annotation/AsyncAnnotationBeanPostProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ApplicationContextAwareProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/AnnotationConfigApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/LocaleContextMessageInterpolator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/AnnotationScopeMetadataResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/StaticMessageSource.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/MBeanServerConnectionFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/context/support/DefaultLifecycleProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/ScheduledTaskRegistrar.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/expression/BeanExpressionContextAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ConfigurationClassParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/CronSequenceGenerator.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/support/FormattingConversionService.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/MBeanExporter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ConfigurationClassMethod.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/number/NumberFormatter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiObjectTargetSource.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationContextException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/InvalidMetadataException.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/notification/UnableToSendNotificationException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/SchedulingAwareRunnable.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/RemoteLookupFailureException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/MBeanServerNotFoundException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/BindingResultUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/EjbAccessException.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/SimpleThrowawayClassLoader.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/MBeanInfoRetrievalException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/RemoteConnectFailureException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/SchedulingException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/MapBindingResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/MBeanExportException.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/glassfish/ClassTransformerAdapter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/UnableToRegisterMBeanException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/JmxException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ResourceLoaderAware.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/EventPublicationInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/BindingErrorProcessor.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/naming/KeyNamingStrategy.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/ResourceOverridingShadowingClassLoader.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/MethodExclusionMBeanInfoAssembler.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/glassfish/GlassFishClassLoaderAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/CommonAnnotationBeanPostProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/ModelMap.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/backportconcurrent/ThreadPoolTaskExecutor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ThreadPoolTaskScheduler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/ScheduledTasksBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/ConnectorServerFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/Phased.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Lazy.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ReloadableResourceBundleMessageSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/SmartLifecycle.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/AbstractRefreshableApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/annotation/AsyncResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ScheduledExecutorFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/AnnotationFormatterFactory.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/timer/ScheduledTimerTask.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ConfigurableApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/ExecutorBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/annotation/AsyncAnnotationAdvisor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/number/AbstractNumberFormatter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/ReadablePartialPrinter.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/SimpleHttpServerFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/DependsOn.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/BeanPropertyBindingResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ExecutorConfigurationSupport.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/SourceFilteringListener.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/JndiRmiProxyFactoryBean.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/JmxUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ImportResource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ConcurrentTaskExecutor.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/ManagedMetric.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ComponentScanBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/SchedulerBeanDefinitionParser.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/JodaTimeContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/timer/TimerTaskExecutor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/AnnotatedBeanDefinitionReader.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ReadOnlySystemAttributesMap.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/NotificationListenerRegistrar.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/SimpleTriggerContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/BeanValidationPostProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/Validator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/MBeanProxyFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/naming/MetadataNamingStrategy.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiTemplate.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/SchedulingTaskExecutor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/weaving/AspectJWeavingEnabler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/ApplicationEventMulticaster.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ScopedProxyMode.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Scope.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/AbstractXmlApplicationContext.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ConcurrentTaskScheduler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/GenericApplicationListenerAdapter.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/BindingResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Jsr330ScopeMetadataResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiProxyFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/backportconcurrent/ConcurrentTaskExecutor.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/Lifecycle.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/support/UiApplicationContextUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedMetric.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/SimpleRemoteStatelessSessionProxyFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/MBeanExportOperations.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/DelegatingErrorHandlingRunnable.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/Trigger.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/support/SimpleJndiBeanFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ClassPathScanningCandidateComponentProvider.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/GenericApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/Model.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ContextTypeMatchClassLoader.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/MessageSourceResourceBundle.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ClassPathBeanDefinitionScanner.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Import.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/annotation/NumberFormat.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/beanvalidation/SpringConstraintValidatorFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/DateFormatter.java
*

/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiServiceExporter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ScheduledExecutorTask.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/TriggerContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/glassfish/GlassFishLoadTimeWeaver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/MethodInvokingRunnable.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/weaving/LoadTimeWeaverAwareProcessor.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/ReadableInstantPrinter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/timer/TimerFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/LifecycleProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/annotation/DateTimeFormat.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/LocalStatelessSessionProxyFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Bean.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/SmartApplicationListener.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/AbstractRefreshableConfigApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationListener.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/FormatterRegistry.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/jboss/JBossLoadTimeWeaver.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/MBeanServerFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/timer/DelegatingTimerTask.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/JmxAttributeSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ScopedProxyCreator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/Printer.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/jmx/export/annotation/AnnotationJmxAttributeSource.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/GenericXmlApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiRegistryFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/JodaTimeConverters.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/access/ContextJndiBeanFactoryLocator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/AnnotationConfigUtils.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/annotation/Async.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/AnnotationBeanNameGenerator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/expression/BeanFactoryAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Configuration.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/timer/MethodInvokingTimerTaskFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/Primary.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/PeriodicTrigger.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/JodaTimeFormattingConfigurer.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ThreadPoolExecutorFactoryBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/support/TaskUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/DateTimeParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/config/TaskNamespaceHandler.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedResource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/datetime/joda/MillisecondInstantPrinter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ReschedulingRunnable.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/context/support/SimpleThreadScope.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/ThreadPoolTaskExecutor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/Formatter.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/format/Parser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/InterfaceBasedMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/SimpleApplicationEventMulticaster.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/interceptor/SpringBeanAutowiringInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ScopeMetadata.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationBasedAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/ThemeSource.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/ContextStartedEvent.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/concurrent/CustomizableThreadFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/LoadTimeWeaverBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/groovy/GroovyObjectCustomizer.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/support/SimpleTheme.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/ManagedResource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiLookupFailureException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/weaving/LoadTimeWeaverAware.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/WeavingTransformer.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/config/AbstractJndiLocatingBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationEvent.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/AbstractSessionBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/config/RemoteStatelessSessionBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationExecutor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/ScriptCompilationException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/stereotype/Component.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ScopeMetadataResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/oc4j/OC4JLoadTimeWeaver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/config/ScriptBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/AbstractMessageDrivenBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/UrlBasedRemoteAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/AbstractJmsMessageDrivenBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/Theme.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/JndiRmiServiceExporter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationResult.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/ValidationUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/ManagedOperationParameter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/ContextStoppedEvent.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/RemoteInvocationFailureException.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/AnnotationMBeanExporter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/AbstractEnterpriseBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/SmartSessionBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/weblogic/WebLogicClassPreProcessorAdapter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/ManagedAttribute.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/ApplicationContextEvent.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/ContextRefreshedEvent.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedNotification.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationContextAware.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedNotifications.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/TypeMismatchNamingException.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/config/JndiLookupBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationBasedExporter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/SimpleInstrumentableClassLoader.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/stereotype/Service.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/AbstractStatefulSessionBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/config/LocalStatelessSessionBeanDefinitionParser.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/SpringModelMBean.java

* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/AbstractJmxAttribute.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/notification/NotificationPublisherAware.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/ScannedGenericBeanDefinition.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/config/JeeNamespaceHandler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiInvocationWrapper.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scheduling/backportconcurrent/CustomizableThreadFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/support/AbstractStatelessSessionBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/naming/IdentityNamingStrategy.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiInvocationHandler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/DefaultRemoteInvocationExecutor.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/ReflectiveLoadTimeWeaver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/config/LangNamespaceHandler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/config/LangNamespaceUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/config/ScriptingDefaultsParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/soap/SoapFaultException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/stereotype/Repository.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/RemoteProxyFailureException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/ManagedNotification.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/SimpleLoadTimeWeaver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/notification/ModelMBeanNotificationPublisher.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/event/ContextClosedEvent.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/stereotype/Controller.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/DefaultRemoteInvocationFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2009 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
* {@link FactoryBean} that obtains a WebSphere {@link javax.management.MBeanServer}
* reference through WebSphere's proprietary <code>AdminServiceFactory</code> API,
* available on WebSphere 5.1 and higher.
*
* <p>Exposes the <code>MBeanServer</code> for bean references.
* This FactoryBean is a direct
alternative to {@link MBeanServerFactoryBean},
* which uses standard JMX 1.2 API to access the platform's MBeanServer.
*
* @author Juergen Hoeller
* @author Rob Harrop
* @since 2.0.3
* @see com.ibm.websphere.management.AdminServiceFactory#getMBeanFactory()
* @see com.ibm.websphere.management.MBeanFactory#getMBeanServer()
* @see javax.management.MBeanServer
* @see MBeanServerFactoryBean
*/

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/WebSphereMBeanServerFactoryBean.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2008 the original author or authors.
*

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/CodebaseAwareObjectInputStream.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/InvalidInvocationException.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/MBeanExporterListener.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/bsh/BshScriptUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/StaticApplicationContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/AbstractSlsbInvokerInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/ShadowingClassLoader.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RemoteInvocationSerializingExporter.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/NotificationListenerBean.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/InvocationFailureException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/ScriptSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/DirectFieldBindingResult.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationTraceInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/jruby/JRubyScriptFactory.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/access/ContextSingletonBeanFactoryLocator.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/validation/support/BindingAwareModelMap.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/BindException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/RemoteAccessException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocationUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/ExtendedModelMap.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/instrument/classloading/LoadTimeWeaver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/support/ResourceScriptSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/AnnotationConfigBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/PropertyOverrideBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiClientInterceptorUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/MBeanExportBeanDefinitionParser.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/AbstractConfigurableMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteExporter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemotingSupport.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/MethodNameBasedMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/support/RefreshableScriptTargetSource.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/PropertyPlaceholderBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/support/StaticScriptSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/i18n/SimpleLocaleContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/FieldError.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/annotation/FilterType.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ApplicationObjectSupport.java
*

/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/DefaultMessageCodesResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/MBeanServerBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/LocalSlsbInvokerInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/MessageSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteAccessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiClientInterceptor.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/jruby/JRubyScriptUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/ConnectorDelegate.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/access/MBeanConnectFailureException.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/DefaultMessageSourceResolvable.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jndi/JndiCallback.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/RmiBasedExporter.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/support/RemoteInvocation.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/groovy/GroovyScriptFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/Errors.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/AbstractPropertyLoadingBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/AbstractErrors.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/SimpleRemoteSlsbInvokerInterceptor.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/NotificationListenerHolder.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/support/ResourceBundleThemeSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/DelegatingMessageSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/ScriptFactory.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/jmx/support/MBeanRegistrationSupport.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/bsh/BshScriptFactory.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/support/ObjectNameManager.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/scripting/support/ScriptFactoryPostProcessor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/ObjectError.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/SpringConfiguredBeanDefinitionParser.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/support/ResourceBundleMessageSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/config/ContextNamespaceHandler.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ejb/access/AbstractRemoteSlsbInvokerInterceptor.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/remoting/rmi/JndiRmiClientInterceptor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/MessageSourceAware.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationEventPublisherAware.java
*
 /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/ui/context/support/DelegatingThemeSource.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-

jar/org/springframework/jmx/export/assembler/SimpleReflectiveMBeanInfoAssembler.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/naming/ObjectNamingStrategy.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/i18n/LocaleContext.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/validation/MessageCodesResolver.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedOperation.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedOperationParameter.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/JmxMetadataUtils.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/metadata/ManagedOperation.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedOperationParameters.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/context/ApplicationEventPublisher.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/assembler/MBeanInfoAssembler.java
*
/opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/annotation/ManagedAttribute.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/naming/SelfNaming.java
* /opt/cola/permits/172894458_1695318099.194561/0/org-springframework-context-sources-3-0-5-release-2-jar/org/springframework/jmx/export/notification/NotificationPublisher.java

1.63 stax-api 1.0.1

1.63.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

// --
// NOTICE file corresponding to the section 4d of The Apache License,

// Version 2.0, in this case for Streaming API for XML (STAX API 1.0)

// --

Streaming API for XML (STAX API 1.0)
Copyright 2003-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache ServiceMix
Copyright 2007-2008 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.64 thrift 0.9.2

1.64.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits
its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use
the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without
limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish
on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to
a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed
under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this

License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables
containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable
source code for the Library including whatever
changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the

Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user
installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the
major
components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people
 have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

 12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

 13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public
 License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status

of
all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line
and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or

Also add information on how to contact you by electronic and paper mail.

You should
also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
Tue Oct 24 12:28:44 CDT 2006

SOFTWARE
OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

----

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with
separate copyright notices and license terms. Your use of the source
code for the these subcomponents is subject to the terms and
conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4
components:

For the compiler/cpp/src/thrift/md5.[ch] components:

/*
 Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

 This software is provided 'as-is', without any express or implied
 warranty. In no event will the authors be held liable for any damages
 arising from the use of this software.

 Permission is granted to anyone to use this software for any purpose,
 including commercial applications, and to alter it and redistribute it
 freely, subject to the following restrictions:

 1. The origin of this software must not be misrepresented; you must not
    claim that you wrote the original software. If you use this software
    in a product, an acknowledgment in the product documentation would be
    appreciated but is not required.
 2. Altered source versions must be plainly marked as such, and must not be
    misrepresented as being the original software.
 3. This notice may not be removed or altered from any source distribution.

 L. Peter Deutsch
 ghost@aladdin.com

*/

For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki,
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components:
    Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser General Public License 2.1
(see doc/lgpl-2.1.txt for the full terms of this license)

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

This package was debianized by Thrift Developer's <dev@thrift.apache.org>.

This package and the Debian packaging is licensed under the Apache License,
see `/usr/share/common-licenses/Apache-2.0'.

The following informations was copied from Apache Thrift LICENSE file.

SOFTWARE DISTRIBUTED WITH THRIFT:

The Apache Thrift software includes a number of subcomponents with
separate copyright notices and license terms. Your use of the source
code for the these subcomponents is subject to the terms and
conditions of the following licenses.

Portions of the following files are licensed under the MIT License:

lib/erl/src/Makefile.am

Please see doc/otp-base-license.txt for the full terms of this license.

The following files contain some portions of code contributed under
the Thrift Software License (see doc/old-thrift-license.txt), and relicensed
under
the Apache 2.0 License:

compiler/cpp/Makefile.am
compiler/cpp/src/generate/t_cocoa_generator.cc
compiler/cpp/src/generate/t_cpp_generator.cc
compiler/cpp/src/generate/t_csharp_generator.cc
compiler/cpp/src/generate/t_erl_generator.cc
compiler/cpp/src/generate/t_hs_generator.cc
compiler/cpp/src/generate/t_java_generator.cc
compiler/cpp/src/generate/t_ocaml_generator.cc
compiler/cpp/src/generate/t_perl_generator.cc
compiler/cpp/src/generate/t_php_generator.cc
compiler/cpp/src/generate/t_py_generator.cc
compiler/cpp/src/generate/t_rb_generator.cc
compiler/cpp/src/generate/t_st_generator.cc
compiler/cpp/src/generate/t_xsd_generator.cc
compiler/cpp/src/main.cc

compiler/cpp/src/parse/t_field.h
compiler/cpp/src/parse/t_program.h
compiler/cpp/src/platform.h
compiler/cpp/src/thriftl.ll
compiler/cpp/src/thrifty.yy
lib/csharp/src/Protocol/TBinaryProtocol.cs
lib/csharp/src/Protocol/TField.cs
lib/csharp/src/Protocol/TList.cs
lib/csharp/src/Protocol/TMap.cs
lib/csharp/src/Protocol/TMessage.cs
lib/csharp/src/Protocol/TMessageType.cs
lib/csharp/src/Protocol/TProtocol.cs
lib/csharp/src/Protocol/TProtocolException.cs
lib/csharp/src/Protocol/TProtocolFactory.cs
lib/csharp/src/Protocol/TProtocolUtil.cs
lib/csharp/src/Protocol/TSet.cs
lib/csharp/src/Protocol/TStruct.cs
lib/csharp/src/Protocol/TType.cs
lib/csharp/src/Server/TServer.cs
lib/csharp/src/Server/TSimpleServer.cs
lib/csharp/src/Server/TThreadPoolServer.cs
lib/csharp/src/TApplicationException.cs
lib/csharp/src/Thrift.csproj
lib/csharp/src/Thrift.sln
lib/csharp/src/TProcessor.cs
lib/csharp/src/Transport/TServerSocket.cs
lib/csharp/src/Transport/TServerTransport.cs
lib/csharp/src/Transport/TSocket.cs
lib/csharp/src/Transport/TStreamTransport.cs
lib/csharp/src/Transport/TTransport.cs
lib/csharp/src/Transport/TTransportException.cs
lib/csharp/src/Transport/TTransportFactory.cs
lib/csharp/ThriftMSBuildTask/Properties/AssemblyInfo.cs
lib/csharp/ThriftMSBuildTask/ThriftBuild.cs
lib/csharp/ThriftMSBuildTask/ThriftMSBuildTask.csproj
lib/rb/lib/thrift.rb
lib/st/README
lib/st/thrift.st
test/OptionalRequiredTest.cpp
test/OptionalRequiredTest.thrift
test/ThriftTest.thrift

For the aclocal/ax_boost_base.m4 and contrib/fb303/aclocal/ax_boost_base.m4 components:

#   modification, are permitted in any medium without royalty provided
#   the copyright notice and this notice are preserved.

For the compiler/cpp/src/md5.[ch] components:

/*
Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use
of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

*/

For the lib/rb/setup.rb: Copyright (c) 2000-2005 Minero Aoki,
lib/ocaml/OCamlMakefile and lib/ocaml/README-OCamlMakefile components:
Copyright (C) 1999 - 2007 Markus Mottl

Licensed under the terms of the GNU Lesser
General Public License 2.1
(see doc/lgpl-2.1.txt for the full terms of this license)
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.
Apache Thrift
Copyright 2006-2010 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.65 zlib 1.2.1

1.66 prop-types 15.6.0

1.66.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.67 commons-codec 1.10

1.67.1 Available under license :

Apache Commons Codec
Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.68 commons-configuration 1.10

1.68.1 Available under license :

Apache Commons Configuration
Copyright 2001-2013 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

1.69 jul-to-slf4j 1.7.5

1.69.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*
*/
```

Found in path(s):

* /opt/cola/permits/173047791_1695320158.037155/0/jul-to-slf4j-1-7-5-sources-jar/org/slf4j/bridge/SLF4JBridgeHandler.java

1.70 rhino 1.7.7.1

1.70.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
/*
        * During the great date rewrite of 1.3, we tried to track the
        * evolving ECMA standard, which then had a definition of
        * getYear which always subtracted 1900.  Which we
        * implemented, not realizing that it was incompatible with
        * the old behavior...  now, rather than thrash the behavior
        * yet again, we've decided to leave it with the - 1900
        * behavior and point people to the getFullYear method.  But
        * we try to protect existing scripts that have specified a
        * version...
        */
```

Found in path(s):

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ContextFactory.java

No license file was found, but licenses were detected in source scan.

```
/* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
```

Found in path(s):

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/OptTransformer.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/ShellConsole.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/CachingModuleScriptProviderBase.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-

zip/org/mozilla/javascript/ScriptStackElement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/SourceReader.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/StackStyle.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/OptRuntime.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/Require.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/annotations/JSGetter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/RequireBuilder.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/ParsedContentType.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/UrlConnectionExpiryCalculator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/ModuleSourceProviderBase.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/SecureCaller.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/ModuleSource.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/ModuleScriptProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/Optimizer.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/ModuleScope.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/OptFunctionNode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XmlNode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/annotations/JSConstructor.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/StrongCachingModuleScriptProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/ModuleSourceProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/UrlConnectionSecurityDomainProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/MultiModuleScriptProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/UrlModuleSourceProvider.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/SoftCachingModuleScriptProvider.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/Block.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/annotations/JSFunction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/jsc/Main.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/annotations/JSSetter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/RhinoSecurityManager.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/annotations/JSStaticFunction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/provider/DefaultUrlConnectionExpiryCalculator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/PolicySecurityController.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/DefiningClassLoader.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/commonjs/module/ModuleScript.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XmlProcessor.java
No license file was found, but licenses were detected in source scan.

/* -*- Mode: java; tab-width: 4; indent-tabs-mode: 1; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/idswitch/SwitchGenerator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/IdScriptableObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/idswitch/CodePrinter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/json/JsonParser.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/idswitch/FileBody.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJSON.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeMath.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/idswitch/Main.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/idswitch/IdValuePair.java

No license file was found, but licenses were detected in source scan.

```
/*
* @(#)TreeTableModelAdapter.java       1.2 98/10/27
*
* Copyright 1997, 1998 Sun Microsystems, Inc.  All Rights Reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Sun Microsystems nor the names of its
*     contributors may be used to endorse or promote products derived
*     from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/treetable/TreeTableModelAdapter.java

No license file was found, but licenses were detected in source scan.

```
/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
```

```
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
// throws a special exception. This ensures execution of all pending
```

Found in path(s):

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeGenerator.java

No license file was found, but licenses were detected in source scan.

```
// Copyright 2011 the V8 project authors. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
//       notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
//       copyright notice, this list of conditions and the following
//       disclaimer in the documentation and/or other materials provided
//     * Neither the name of Google Inc. nor the names of its
//       from this software without specific prior written permission.
```

Found in path(s):

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/v8dtoa/DoubleConversion.java

No license file was found, but licenses were detected in source scan.

```
/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
/**
* This is interface defines a protocol for the reporting of
* errors during JavaScript translation in IDE-mode.
* If the {@link org.mozilla.javascript.Parser}'s error reporter is
* set to an instance of this interface, then this interface's
* {@link #warning} and {@link #error} methods are called instead
* of the {@link org.mozilla.javascript.ErrorReporter} versions. <p>
*
* These methods take a source char offset and a length.  The
* rationale is that in interactive IDE-type environments, the source
* is available and the IDE will want to indicate where the error
* occurred and how much code participates in it.  The start
 and length
* are generally chosen to fit within a single line, for readability,
* but the client is free to use the AST to determine the affected
* node(s) from the start position and change the error or warning's
```

* display bounds.<p>
*
* @author Steve Yegge
*/

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/IdeErrorReporter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/treetable/AbstractCellEditor.java
No license file was found, but licenses were detected in source scan.

```
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
//       notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
//       copyright notice, this list of conditions and the following
//       disclaimer in the documentation and/or other materials provided
//     * Neither the name of Google Inc. nor the names of its
//       from this software without specific prior written permission.
```

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/v8dtoa/DoubleHelper.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/v8dtoa/FastDtoa.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/v8dtoa/DiyFp.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/v8dtoa/CachedPowers.java

No license file was found, but licenses were detected in source scan.

```
/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
```

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Comment.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeInt32Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/classfile/ByteCode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeFloat64Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeInt8Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ClassCache.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/SwingGui.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeCallSite.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaTopPackage.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/QName.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Interpreter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/jdk13/VMBridge_jdk13.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/serialize/ScriptableOutputStream.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaArray.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ContextListener.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/GeneratorExpressionLoop.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Delegator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/PropertyGet.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeArrayBufferView.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeUint16Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ConditionalExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlDotQuery.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaClass.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/JavaPolicySecurity.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeTypedArrayIterator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ArrayLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/regexp/NativeRegExp.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/IRFactory.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/Main.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/Main.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-

zip/org/mozilla/javascript/v8dtoa/FastDtoaBuilder.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ClassShutter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Callable.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/InterpreterData.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaMethod.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/FunctionCall.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/ConsoleTextArea.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Icode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/GeneratorExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/NewExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/TokenStream.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/classfile/ClassFileWriter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeCall.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeGlobal.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlMemberGet.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/WrappedException.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/VariableInitializer.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/CompilerEnvirons.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XMLCtor.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/KeywordLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Wrapper.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ObjectProperty.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XMLName.java
*

/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Name.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/UintMap.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ArrayComprehension.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Kit.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/BoundFunction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/MemberBox.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeContinuation.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/NumberLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Parser.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/debug/DebuggableObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeUint8ClampedArray.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Loop.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/JSConsole.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlString.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/UnaryExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ElementGet.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XMLList.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Script.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/regexp/SubString.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XMLObjectImpl.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/regexp/NativeRegExpCtor.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ImporterTopLevel.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/UniqueTag.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Scriptable.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-

zip/org/mozilla/javascript/LazilyLoadedCtor.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/LetNode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xml/XMLLib.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/CodeGenerator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/SecurityUtilities.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/SwitchStatement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeTypedArrayView.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/LabeledStatement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xml/XMLObject.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/JavaMembers.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/jdk15/VMBridge_jdk15.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/ScopeProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/DefaultErrorReporter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeError.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/EvaluatorException.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/Conversions.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ScriptNode.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeArrayBuffer.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/SecurityProxy.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlFragment.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ForInLoop.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ErrorCollector.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ConstProperties.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeDataView.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/regexp/RegExpImpl.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/IdFunctionObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Label.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Symbol.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/FunctionNode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/Environment.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/RegExpProxy.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ExpressionStatement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/EmptyStatement.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeScript.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ContextAction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/GeneratedClassLoader.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeString.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ObjToIntMap.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeDate.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Ref.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Yield.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Arguments.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/CatchClause.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ForLoop.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlPropRef.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/Codegen.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/EcmaError.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/SpecialRef.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/ByteIo.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ConsString.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlElemRef.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/Dim.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/InterfaceAdapter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Node.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/TopLevel.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Context.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/JavaAdapter.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/NodeVisitor.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/RhinoException.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ScriptableObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/SourceProvider.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/BreakStatement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/VMBridge.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Undefined.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeWith.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/serialize/ScriptableInputStream.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NodeTransformer.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/InfixExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/debug/DebugFrame.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/RegExpLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Scope.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/RefCallable.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Function.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ObjectLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/JavaScriptException.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/AstRoot.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/Namespace.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeInt16Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/BaseFunction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ErrorNode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Jump.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ContinueStatement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Decompiler.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ArrayComprehensionLoop.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XML.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/WrapFactory.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/ShellContextFactory.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeFloat32Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ParenthesizedExpression.java
*

/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeIterator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XMLLibImpl.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/DoLoop.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlRef.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Evaluator.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Assignment.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaConstructor.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/SecurityController.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/debug/DebuggableScript.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/VariableDeclaration.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/WhileLoop.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/XmlExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ObjArray.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/TryStatement.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ContinuationPending.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/EmptyExpression.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/IfStatement.java
*
/opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/IdFunctionCall.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ExternalArrayData.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/xmlimpl/XMLWithScope.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ParseProblem.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/debug/Debugger.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeArray.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/InterpretedFunction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeBoolean.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/GuiCallback.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeUint8Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/AstNode.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/Global.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/SwitchCase.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/FunctionObject.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/ToolErrorReporter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ScriptRuntime.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/DestructuringForm.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/StringLiteral.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/QuitAction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeFunction.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeNumber.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/Block.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ErrorReporter.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ThrowStatement.java
*
 /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Synchronizer.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/typedarrays/NativeUint32Array.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/Token.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/optimizer/ClassCompiler.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/WithStatement.java

* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/shell/ShellLine.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/NativeJavaPackage.java
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/ast/ReturnStatement.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/treetable/JTreeTable.java
No license file was found, but licenses were detected in source scan.

/* -*- Mode: java; tab-width: 8; indent-tabs-mode: nil; c-basic-offset: 4 -*-

```
*
* This Source Code Form is subject to the terms of the Mozilla Public
* License, v. 2.0. If a copy of the MPL was not distributed with this
* file, You can obtain one at http://mozilla.org/MPL/2.0/. */
/****************************************************************
 *
 * The author of this software is David M. Gay.
 *
 * Copyright (c) 1991, 2000, 2001 by Lucent Technologies.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose without fee is hereby granted, provided that this entire notice
 * is included in all copies of any software which is or includes a copy
 * or modification of this software and in all copies of the supporting
 * documentation for such software.
 *
 * THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
 * WARRANTY.  IN PARTICULAR, NEITHER THE AUTHOR
 NOR LUCENT MAKES ANY
 * REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
 * OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
 *
 ***************************************************************/
```

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/DToA.java
No license file was found, but licenses were detected in source scan.

```
/*
* TreeTableModel.java
*
* Copyright 1998 Sun Microsystems, Inc.  All Rights Reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Sun Microsystems nor the names of its
*     contributors may be used to endorse or promote products derived
*     from this software without specific prior written permission.
*
```

Found in path(s):
* /opt/cola/permits/173065398_1695320250.7149785/0/rhino-1-7-7-1-sources-zip/org/mozilla/javascript/tools/debugger/treetable/TreeTableModel.java

1.71 snake-yaml 1.12

1.72 xpp 1.1.4c

1.72.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
*    license;
*
* 2) All redistributions in binary form must reproduce the above
*    copyright notice, this list of conditions and the disclaimer
*    listed in this
 license in the documentation and/or other
*    materials provided with the distribution;
*
* 3) Any documentation included with all redistributions must include
*    the following acknowledgement:
*
*      "This product includes software developed by the Indiana
*      University Extreme! Lab.  For further information please visit
*      http://www.extreme.indiana.edu/"
*
*    Alternatively, this acknowledgment may appear in the software
*    itself, and wherever such third-party acknowledgments normally
*    appear.
*
* 4) The name "Indiana University" or "Indiana University
*    Extreme! Lab" shall not be used to endorse or promote
*    products derived from this software without prior written
*    permission from Indiana University.  For written permission,
*    please contact http://www.extreme.indiana.edu/.
*
* 5) Products derived from this software may not use "Indiana
*    University" name nor may "Indiana University" appear in their name,
*    without
 prior written permission of the Indiana University.
*
* Indiana University provides no reassurances that the source code
* provided does not infringe the patent or any other intellectual
* property rights of any other entity.  Indiana University disclaims any
* liability to any recipient for claims brought by any other entity
* based on infringement of intellectual property rights or otherwise.
*
* LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH
* NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA
* UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT
* SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR
* OTHER PROPRIETARY RIGHTS.  INDIANA UNIVERSITY MAKES NO WARRANTIES THAT
* SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP
* DOORS", "WORMS", OR OTHER HARMFUL CODE.  LICENSEE ASSUMES THE ENTIRE
* RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS,
* AND TO THE PERFORMANCE AND VALIDITY
 OF INFORMATION GENERATED USING
* SOFTWARE.
*/
```

Found in path(s):

* /opt/cola/permits/173080559_1695320166.6345239/0/xpp3-1-1-4c-sources-zip/org/xmlpull/mxp1/MXParserNonValidating.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2002-2004 Extreme! Lab, Indiana University. All rights reserved.
*
* This software is open source. See the bottom of this file for the licence.
*
* $Id: MXParserCachingStrings.java,v 1.11 2006/10/23 13:36:27 aslom Exp $
*/
/*
* Indiana University Extreme! Lab Software License, Version 1.2
*
* Copyright (c) 2002-2004 The Trustees of Indiana University.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
* 1) All redistributions of source code must retain the above
*    copyright notice, the list of authors in the original source
*    code, this list of conditions and the disclaimer listed in this
*    license;
*
* 2) All redistributions in binary form must reproduce the above
*    copyright notice, this list of conditions and the disclaimer
*    listed in
 this license in the documentation and/or other
*    materials provided with the distribution;
*
* 3) Any documentation included with all redistributions must include
*    the following acknowledgement:
*
*      "This product includes software developed by the Indiana
*      University Extreme! Lab.  For further information please visit
*      http://www.extreme.indiana.edu/"
*
*    Alternatively, this acknowledgment may appear in the software
*    itself, and wherever such third-party acknowledgments normally
*    appear.
*
* 4) The name "Indiana University" or "Indiana University
*    Extreme! Lab" shall not be used to endorse or promote
*    products derived from this software without prior written
```

*   permission from Indiana University. For written permission,
*   please contact http://www.extreme.indiana.edu/.
*
* 5) Products derived from this software may not use "Indiana
*   University" name nor may "Indiana University" appear in their name,
*   without
 prior written permission of the Indiana University.
*
* Indiana University provides no reassurances that the source code
* provided does not infringe the patent or any other intellectual
* property rights of any other entity. Indiana University disclaims any
* liability to any recipient for claims brought by any other entity
* based on infringement of intellectual property rights or otherwise.
*
* LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH
* NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA
* UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT
* SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR
* OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT
* SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP
* DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE
* RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS,
* AND TO THE PERFORMANCE AND VALIDITY
 OF INFORMATION GENERATED USING
* SOFTWARE.
*/

Found in path(s):
* /opt/cola/permits/173080559_1695320166.6345239/0/xpp3-1-1-4c-sources-zip/org/xmlpull/mxp1/MXParserCachingStrings.java
No license file was found, but licenses were detected in source scan.

/*
* The Apache Software License, Version 1.1
*
*
* Copyright (c) 2001-2003 The Apache Software Foundation. All rights
* reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in

Found in path(s):
* /opt/cola/permits/173080559_1695320166.6345239/0/xpp3-1-1-4c-sources-zip/javax/xml/namespace/QName.java
No license file was found, but licenses were detected in source scan.

```
* This software is open source. See the bottom of this file for the license.
*
* $Id: MXParser.java,v 1.52 2006/11/09 18:29:37 aslom Exp $
*/
/*
* Indiana University Extreme! Lab Software License, Version 1.2
*
* Copyright (C) 2003 The Trustees of Indiana University.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
* 1) All redistributions of source code must retain the above
*    copyright notice, the list of authors in the original source
*    code, this list of conditions and the disclaimer listed in this
*    license;
*
* 2) All redistributions in binary form must reproduce the above
*    copyright notice, this list of conditions and the disclaimer
*    listed in this license in the documentation
 and/or other
*    materials provided with the distribution;
*
* 3) Any documentation included with all redistributions must include
*    the following acknowledgement:
*
*      "This product includes software developed by the Indiana
*      University Extreme! Lab.  For further information please visit
*      http://www.extreme.indiana.edu/"
*
*    Alternatively, this acknowledgment may appear in the software
*    itself, and wherever such third-party acknowledgments normally
*    appear.
*
* 4) The name "Indiana University" or "Indiana University
*    Extreme! Lab" shall not be used to endorse or promote
*    products derived from this software without prior written
*    permission from Indiana University.  For written permission,
*    please contact http://www.extreme.indiana.edu/.
*
* 5) Products derived from this software may not use "Indiana
*    University" name nor may "Indiana University" appear in their name,
*    without prior written permission
 of the Indiana University.
*
* Indiana University provides no reassurances that the source code
* provided does not infringe the patent or any other intellectual
* property rights of any other entity.  Indiana University disclaims any
* liability to any recipient for claims brought by any other entity
* based on infringement of intellectual property rights or otherwise.
*
* LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH
* NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA
* UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT
* SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR
* OTHER PROPRIETARY RIGHTS.  INDIANA UNIVERSITY MAKES NO WARRANTIES THAT
* SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP
* DOORS", "WORMS", OR OTHER HARMFUL CODE.  LICENSEE ASSUMES THE ENTIRE
* RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS,
* AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED
 USING
* SOFTWARE.
*/
```

Found in path(s):

* /opt/cola/permits/173080559_1695320166.6345239/0/xpp3-1-1-4c-sources-zip/org/xmlpull/mxp1/MXParser.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2002-2004 Extreme! Lab, Indiana University. All rights reserved.
*
* This software is open source. See the bottom of this file for the licence.
*
* $Id: MXParserFactory.java,v 1.6 2004/03/02 09:14:41 aslom Exp $
*/
/*
* Indiana University Extreme! Lab Software License, Version 1.2
*
* Copyright (c) 2002-2004 The Trustees of Indiana University.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
* 1) All redistributions of source code must retain the above
*    copyright notice, the list of authors in the original source
*    code, this list of conditions and the disclaimer listed in this
*    license;
*
* 2) All redistributions in binary form must reproduce the above
*    copyright notice, this list of conditions and the disclaimer
*    listed in this license
```

```
 in the documentation and/or other
*    materials provided with the distribution;
*
* 3) Any documentation included with all redistributions must include
*    the following acknowledgement:
*
*      "This product includes software developed by the Indiana
*      University Extreme! Lab.  For further information please visit
*      http://www.extreme.indiana.edu/"
*
*    Alternatively, this acknowledgment may appear in the software
*    itself, and wherever such third-party acknowledgments normally
*    appear.
*
* 4) The name "Indiana University" or "Indiana University
*    Extreme! Lab" shall not be used to endorse or promote
*    products derived from this software without prior written
*    permission from Indiana University.  For written permission,
*    please contact http://www.extreme.indiana.edu/.
*
* 5) Products derived from this software may not use "Indiana
*    University" name nor may "Indiana University" appear in their name,
*    without prior
 written permission of the Indiana University.
*
* Indiana University provides no reassurances that the source code
* provided does not infringe the patent or any other intellectual
* property rights of any other entity.  Indiana University disclaims any
* liability to any recipient for claims brought by any other entity
* based on infringement of intellectual property rights or otherwise.
*
* LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH
* NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA
* UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT
* SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR
* OTHER PROPRIETARY RIGHTS.  INDIANA UNIVERSITY MAKES NO WARRANTIES THAT
* SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP
* DOORS", "WORMS", OR OTHER HARMFUL CODE.  LICENSEE ASSUMES THE ENTIRE
* RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS,
* AND TO THE PERFORMANCE AND VALIDITY OF
 INFORMATION GENERATED USING
* SOFTWARE.
*/
```

Found in path(s):

* /opt/cola/permits/173080559_1695320166.6345239/0/xpp3-1-1-4c-sources-zip/org/xmlpull/mxp1/MXParserFactory.java

1.73 avro 1.8.2

1.73.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Avro
Copyright 2009-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.74 httpasyncclient 4.1.2

1.74.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

1.75 javassist 3.18.2

1.75.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/StringMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ShortMemberValue.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/FloatMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/BooleanMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ClassMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/IntegerMemberValue.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ByteMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/CharMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/EnumMemberValue.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/MemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/ArrayMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/AnnotationMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/DoubleMemberValue.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/Annotation.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/MemberValueVisitor.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/LongMemberValue.java
No license file was found, but licenses were detected in source scan.

```
/*
* Javassist, a Java-bytecode translator toolkit.
* Copyright (C) 1999- Shigeru Chiba. All Rights Reserved.
*
* The contents of this file are subject to the Mozilla Public License Version
* 1.1 (the "License"); you may not use this file except in compliance with
* the License.  Alternatively, the contents of this file may be used under
* the terms of the GNU Lesser General Public License Version 2.1 or later,
* or the Apache License Version 2.0.
*
* Software distributed under the License is distributed on an "AS IS" basis,
* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
* for the specific language governing rights and limitations under the
* License.
*/
```

Found in path(s):
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ClassFileWriter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/Cast.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/MethodFilter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/SyntheticAttribute.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/ScopedClassPool.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtBehavior.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/CannotInvokeException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/TypeChecker.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/AssignExpr.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/ConstructorCall.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/Compiler.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/Proxy.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ExceptionTable.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/AppletServer.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/DoubleConst.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/LoaderClassPath.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/CodeGen.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Analyzer.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/ClassPoolTail.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/MethodHandler.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/Dump.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/EnclosingMethodAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ClassFilePrinter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/runtime/Desc.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CannotCompileException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/SignatureAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/ArrayInit.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/CannotReflectException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-

jar/javassist/tools/reflect/Reflection.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/Instanceof.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/SoftValueHashMap.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/runtime/Inner.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/CastExpr.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/StubGenerator.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformReadField.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtArray.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/MultiType.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/ObjectImporter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/ProxyObjectInputStream.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/Mnemonic.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/LocalVariableTypeAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/DuplicateMemberException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/FieldInfo.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/Parser.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/RemoteRef.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/RemoteException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/Loader.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/KeywordTable.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/ScopedClassPoolRepositoryImpl.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewConstructor.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/AnnotationsWriter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/CodeIterator.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/Opcode.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/DeprecatedAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/Loader.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/ClassPool.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CodeConverter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/NoSuchClassError.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtClassType.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/JvstCodeGen.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/ScopedClassPoolFactoryImpl.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/HotSwapper.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ProceedHandler.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/ProxyObject.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/SecurityActions.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Declarator.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformNew.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/ByteArrayClassPath.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ConstPool.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/StackMap.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/Handler.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Subroutine.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Type.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/ClassMap.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/FactoryHelper.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/runtime/Cflow.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/Metalevel.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/CondExpr.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/SubroutineScanner.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtClass.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/IntConst.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/ASTree.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewWrappedConstructor.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/CannotCreateException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewClass.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/stackmap/Tracer.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/web/Webserver.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/Descriptor.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/BadBytecode.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Pair.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/URLClassPath.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ConstantAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/SymbolTable.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/Translator.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/AnnotationDefaultAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Variable.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/NewExpr.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/TokenId.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Executor.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformAccessArrayField.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/framedump.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/ExprEditor.java
*

/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/ControlFlow.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtMember.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/SerializedProxy.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/MultiArrayType.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/NoFieldException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Frame.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/SyntaxError.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Expr.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/Transformer.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/Modifier.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/Lex.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformCall.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/ASTList.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/LineNumberAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/web/Viewer.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/RuntimeSupport.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/stackmap/BasicBlock.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ParameterAnnotationsAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformWriteField.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/ScopedClassPoolRepository.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtConstructor.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/stackmap/TypeTag.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/annotation/AnnotationImpl.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/Sample.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/MemberCodeGen.java

*

/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/FieldAccess.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/LocalVariableAttribute.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/InstructionPrinter.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/CodeAttribute.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/ClassMetaobject.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/AttributeInfo.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/Metaobject.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/MethodDecl.java

*

/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformBefore.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformAfter.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewMethod.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/ClassClassPath.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ByteArray.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ClassFile.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/StackMapTable.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/rmi/ObjectNotFoundException.java

*

/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/InnerClassesAttribute.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/SerialVersionUID.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/IntQueue.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/stackmap/TypedBlock.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/CompileError.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Visitor.java

* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-

jar/javassist/compiler/AccessorMaker.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/LongVector.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewWrappedMethod.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtPrimitiveType.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/ProxyFactory.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/AccessFlag.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformNewClass.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/Bytecode.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/convert/TransformFieldAccess.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/ClassPath.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/BinExpr.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/JvstTypeChecker.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtMethod.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/NotFoundException.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/web/BadHttpRequest.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ByteStream.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/AnnotationsAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/SourceFileAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Keyword.java
*
/opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/runtime/DotClass.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/StringL.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtField.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/MethodCall.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/MethodInfo.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-

jar/javassist/compiler/ast/NewExpr.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Stmnt.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/ProxyObjectOutputStream.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/MemberResolver.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Member.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/FramePrinter.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/Expr.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/tools/reflect/Sample.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/ExceptionsAttribute.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/Symbol.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/util/proxy/Proxy.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/CtNewNestedClass.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/stackmap/MapMaker.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/stackmap/TypeData.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/analysis/Util.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/InstanceOfExpr.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/bytecode/CodeAnalyzer.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/Javac.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/scopedpool/ScopedClassPoolFactory.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/FieldDecl.java
*
 /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/expr/NewArray.java
* /opt/cola/permits/173587136_1695321407.7992156/0/javassist-3-18-2-ga-sources-jar/javassist/compiler/ast/CallExpr.java

1.76 tiger-types 1.4

1.76.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173597647_1695324549.754479/0/tiger-types-1-4-sources-jar/org/jvnet/tiger_types/GenericArrayTypeImpl.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/173597647_1695324549.754479/0/tiger-types-1-4-sources-jar/org/jvnet/tiger_types/Lister.java
* /opt/cola/permits/173597647_1695324549.754479/0/tiger-types-1-4-sources-jar/org/jvnet/tiger_types/Types.java
* /opt/cola/permits/173597647_1695324549.754479/0/tiger-types-1-4-sources-jar/org/jvnet/tiger_types/ParameterizedTypeImpl.java
* /opt/cola/permits/173597647_1695324549.754479/0/tiger-types-1-4-sources-jar/org/jvnet/tiger_types/TypeVisitor.java
* /opt/cola/permits/173597647_1695324549.754479/0/tiger-types-1-4-sources-jar/org/jvnet/tiger_types/WildcardTypeImpl.java

1.77 bunyan 1.8.12

1.77.1 Available under license :

1.78 commons-codec 1.4

1.78.1 Available under license :

----

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.sourceforge.net/test/batch0.tab.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.79 httpcomponents-client 4.5.3

1.79.1 Available under license :

Apache HttpClient
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.80 httpcomponents-core 4.4.6

1.80.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

1.81 javax-ws-rs-api 2.0.1

1.81.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright (C) 2006 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
```

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/GenericEntity.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/ParamConverterProvider.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ContainerResponseContext.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/Priorities.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/ResponseProcessingException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/NoContentException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/Client.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/ClientBuilder.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/InvocationCallback.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/WriterInterceptorContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Form.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/WebTarget.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/package-info.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Configurable.java
*
/opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/SyncInvoker.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/AsyncInvoker.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/ReaderInterceptorContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/Invocation.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ProcessingException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ServerErrorException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/InternalServerErrorException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/NotAuthorizedException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/ClientResponseContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ServiceUnavailableException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/ClientRequestContext.java

* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Configuration.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/RedirectionException.java
*
 /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ForbiddenException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/AbstractMultivaluedMap.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ContainerRequestContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/PreMatching.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ContainerResponseFilter.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/ClientRequestFilter.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/NotSupportedException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/NotAcceptableException.java
*
 /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ContainerRequestFilter.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/NotFoundException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/ClientResponseFilter.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/AsyncResponse.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/NotAllowedException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/BadRequestException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ClientErrorException.java
No license file was found, but licenses were detected in source scan.

* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Link.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/Entity.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ResourceContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/TimeoutHandler.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/FeatureContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ConnectionCallback.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/RuntimeType.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/BeanParam.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ConstrainedTo.java

* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/Suspended.java
*
/opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/package-info.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/CompletionCallback.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/WebApplicationException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Response.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/MediaType.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/CacheControl.java
No license file was found, but licenses were detected in source scan.

* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/ResourceInfo.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/NameBinding.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Feature.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/container/DynamicFeature.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/ReaderInterceptor.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/InterceptorContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/WriterInterceptor.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/MultivaluedHashMap.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/SecurityContext.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/UriBuilderException.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/ContextResolver.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Context.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/Path.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/GenericType.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/StreamingOutput.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/package-info.java
*
/opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/package-info.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/Provider.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/MessageBodyWriter.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/ExceptionMapper.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-

zip/javax/ws/rs/HttpMethod.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/Encoded.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/PathSegment.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/HeaderParam.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/MultivaluedMap.java
*
/opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/Providers.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/CookieParam.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/client/FactoryFinder.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/Consumes.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Cookie.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/Produces.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/OPTIONS.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/NewCookie.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/FactoryFinder.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/PathParam.java
*
 /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/PUT.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Variant.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Application.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/UriInfo.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/UriBuilder.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/MessageBodyReader.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/FormParam.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ApplicationPath.java
*
 /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/MatrixParam.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/GET.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/RuntimeDelegate.java

* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/DefaultValue.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/DELETE.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/HttpHeaders.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/QueryParam.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/HEAD.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/Request.java
*
 /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/core/EntityTag.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/package-info.java
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/POST.java

No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):

```
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
/**
    * Convert the supplied value to a String.
    * <p>
    * This method is reserved for future use. Proprietary JAX-RS extensions may leverage the method.
    * Users should be aware that any such support for the method comes at the expense of producing
    * non-portable code.
    * </p>
    *
    * @param value the value of type {@code T}.
    * @return a String representation of the value.
    *
    * @throws IllegalArgumentException if the supplied object cannot be
    *                                  serialized or is {@code null}.
    */
```

Found in path(s):
* /opt/cola/permits/173977972_1695321855.1381195/0/javax-ws-rs-api-2-0-1-sources-zip/javax/ws/rs/ext/ParamConverter.java

1.82 reflections 0.9.10

1.83 swagger-jaxrs 1.5.0

1.83.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright 2015 SmartBear Software
* <p/>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p/>
```

* http://www.apache.org/licenses/LICENSE-2.0
* <p/>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/173990353_1695321903.8829024/0/swagger-jaxrs-1-5-0-sources-jar/io/swagger/jaxrs/Reader.java
* /opt/cola/permits/173990353_1695321903.8829024/0/swagger-jaxrs-1-5-0-sources-jar/io/swagger/jaxrs/json/JacksonJsonProvider.java
* /opt/cola/permits/173990353_1695321903.8829024/0/swagger-jaxrs-1-5-0-sources-jar/io/swagger/jaxrs/config/ReaderListener.java

1.84 swagger-jersey2-jaxrs 1.5.0

1.85 antlr 3.1.3

1.85.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2007 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

```
group ASTTreeParser;

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
{
   <ASTLabelType> root_<treeLevel> = [treeAdapator newEmptyTree];
   <root:element()>
   <actionsAfterRoot:element()>
   <if(nullableChildList)>
   if ( [input LA:1] == ANTLRTokenTypeDOWN ) {
       [self match:input tokenType:ANTLRTokenTypeDOWN follow:nil]; <checkRuleBacktrackFailure()>
        <children:element()>
       [self match:input tokenType:ANTLRTokenTypeUP follow:nil]; <checkRuleBacktrackFailure()>
   }
   <else>
   [self match:input tokenType:ANTLRTokenTypeDOWN follow:nil]; <checkRuleBacktrackFailure()>
   <children:element()>
   [self match:input tokenType:ANTLRTokenTypeUP follow:nil]; <checkRuleBacktrackFailure()>
   <endif>
   [root_<treeLevel> release];
}<\n>
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(rewriteMode)>retval.tree = (<ASTLabelType>)retval.start;<endif>
>>

// TOKEN AST STUFF

/** ID auto construct */
tokenRef(token,label,elementIndex) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
```

```
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>adaptor.addChild(root_<treeLevel>, <label>.getTree());
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope)
 ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(), root_<treeLevel>);
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
```

<ruleRefRuleRoot(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** Streams for token refs are tree nodes now; override to
* change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
#error Heterogeneous tree support not implemented.
<if(hetero)>
new <hetero>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ObjC/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/serialize.g
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/ST.stg
No license file was found, but licenses were detected in source scan.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/

/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/
group ASTParser;

@rule.setErrorReturnValue()
 ::= <<
retval.tree = (<ASTLabelType>)(ADAPTOR->errorNode(ADAPTOR, INPUT, retval.start, LT(-1), EXCEPTION));
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
ADAPTOR->addChild(ADAPTOR, root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_0));
<if(backtracking)>}<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>



/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>



/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,hetero,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
```

```
//
 rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
<super.matchSet(..., postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )
<endif>ADAPTOR->addChild(ADAPTOR, root_0, <createNodeFromToken(...)>);})>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<if(label)>
<label>=(<labelType>)LT(1);<\n>
<endif>
<super.matchSet(..., postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0
= (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <createNodeFromToken(...)>, root_0));})>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope)
 ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>ADAPTOR->addChild(ADAPTOR, root_0,
<label>.tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0 = (<ASTLabelType>)(ADAPTOR-
>becomeRoot(ADAPTOR, <label>.tree, root_0));
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
```

```
<ruleRef(...)>
<listLabelAST(...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabelAST(...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabelAST(...)>
>>

// WILDCARD AST

wildcard(label,elementIndex)
 ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
ADAPTOR->addChild(ADAPTOR, root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>));
root_0 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_0));
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,hetero) ::= <<
<if(hetero)>
<hetero>New(<label>) <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType>)(ADAPTOR->create(ADAPTOR, <label>))
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp(...)>
<if(backtracking)>
```

```
if ( <actions.(actionScope).synpredgate> )
{<\n>
<endif>
<if(!ruleDescriptor.isSynPred)>
 retval.stop = LT(-1);<\n>
<endif>
 retval.tree
 = (<ASTLabelType>)(ADAPTOR->rulePostProcessing(ADAPTOR, root_0));
 ADAPTOR->setTokenBoundaries(ADAPTOR, retval.tree, retval.start, retval.stop);
   <ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
   <ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<if(backtracking)>
}<\n>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/C/ASTParser.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/CTarget.java
No license file was found, but licenses were detected in source scan.

```
group Perl5 implements ANTLRCore;

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp,
 trace,
    scopes, superClass, literals) ::=
<<
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>

<@imports>
<if(TREE_PARSER)>
<endif>
<if(backtracking)>
<endif>
<@end>

<docComment>
<recognizer>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="ANTLR::Runtime::Lexer")  ::= <<
package <name>;
use ANTLR::Runtime::Class;

use Carp;
use English qw( -no_match_vars ) ;
use Readonly;
use Switch;

use ANTLR::Runtime::BaseRecognizer;
use ANTLR::Runtime::DFA;
use ANTLR::Runtime::NoViableAltException;

extends 'ANTLR::Runtime::Lexer';

use constant {
   HIDDEN => ANTLR::Runtime::BaseRecognizer->HIDDEN
};

use constant {
   <tokens:{ <it.name> => <it.type>, }; separator="\n">
};
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<actions.lexer.members>

sub BUILD {
   my ($self, $arg_ref) = @_;

   $self->init_dfas();
```

```
}

sub get_grammar_file_name {
   return
 "<fileName>";
}

<if(filterMode)>
<filteringNextToken()>
<endif>
<rules; separator="\n\n">

<synpreds:{p | <lexerSynpred(p)>}>

<cyclicDFAs:{dfa | has 'dfa<dfa.decisionNumber>';}; separator="\n">

sub init_dfas {
   my ($self) = @_;

   <cyclicDFAs:{dfa |
   $self->dfa<dfa.decisionNumber>(<name>::DFA<dfa.decisionNumber>->new({ recognizer => $self }));
   }; separator="\n">

   return;
}

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

1;

>>

perlTypeInitMap ::= [
 "$":"undef",
 "@":"()",
 "%":"()",
 default:"undef"
]

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public Token nextToken() {
   while (true) {
      if ( input.LA(1)==CharStream.EOF ) {
          return
 Token.EOF_TOKEN;
      }
      token = null;
 channel = Token.DEFAULT_CHANNEL;
      tokenStartCharIndex = input.index();
      tokenStartCharPositionInLine = input.getCharPositionInLine();
      tokenStartLine = input.getLine();
 text = null;
      try {
          int m = input.mark();
          backtracking=1; <! means we won't throw slow exception !>
          failed=false;
          mTokens();
          backtracking=0;
          <! mTokens backtracks with synpred at backtracking==2
             and we set the synpredgate to allow actions at level 1. !>
          if ( failed ) {
              input.rewind(m);
              input.consume(); <! advance one char and try again !>
          }
          else {
              emit();
              return token;
          }
      }
      catch (RecognitionException re) {
          // shouldn't happen in backtracking mode, but...
          reportError(re);
          recover(re);

  }
   }
}

public void memoize(IntStream input,
  int ruleIndex,
  int ruleStartIndex)
{
if ( backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
if ( backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>
```

```
actionGate() ::= "$self->state->backtracking==0"

filteringActionGate() ::= "backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
        bitsets, inputStreamType, superClass, filterMode,
        ASTLabelType="Object", labelType, members) ::= <<
package <name>;
use ANTLR::Runtime::Class;

use English qw( -no_match_vars ) ;
use Readonly;
use Switch;
use Carp;
use ANTLR::Runtime::BitSet;

extends '<@superClassName><superClass><@end>';

Readonly my $token_names => [
    "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
];

use constant {
<tokens:{ <it.name> => <it.type>,
 }; separator="\n">
};

<bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                    words64=it.bits)>

<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
<@members>
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>

sub BUILD {
    my ($self, $arg_ref) = @_;

<if(backtracking)>
    $self->state->rule_memo({});<\n>
<endif>
}
<@end>

sub get_token_names {
    return $token_names;
}
```

```
sub get_grammar_file_name {
  return "<fileName>";
}

<members>

<rules; separator="\n\n">

<synpreds:{p | <synpred(p)>}>

<cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = __PACKAGE__::DFA<dfa.decisionNumber>->new($self);};
separator="\n">
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>



1;
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType,
superClass="ANTLR::Runtime::Parser", labelType="ANTLR::Runtime::Token",
members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ANTLR::Runtime::TokenStream",
 ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="Object", superClass="ANTLR::Runtime::TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="TreeNodeStream", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start <ruleName>
sub <ruleName>_fragment {
# <ruleDescriptor.parameterScope:parameterScope(scope=it)>
<if(trace)>
     $self->traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
    eval {
        <block>
    };
    $self->traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
    if ($EVAL_ERROR) {
        croak $EVAL_ERROR;
    }
<else>
    <block>
<endif>
}
# $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public final boolean <name>() {
    backtracking++;
    <@start()>
    int start = input.mark();
    try {
        <name>_fragment(); // can never throw exception
    } catch (RecognitionException re) {
        System.err.println("impossible: "+re);
    }
    boolean success = !failed;
    input.rewind(start);
    <@stop()>
    backtracking--;
    failed=false;
    return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>; }
<endif>
>>

/** How to test for failure and return from rule
```

```
 */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>
if ($self->state->failed) {
   return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (backtracking>0) {failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

# $ANTLR start <ruleName>
# <fileName>:<description>
sub <ruleName>() {
   my ($self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>) = @_;
   <if(trace)>$self->traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>

  eval {
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
   };
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   my $exception = $EVAL_ERROR;
   if (ref $exception && $exception->isa('ANTLR::Runtime::RecognitionException')) {
       $self->report_error($exception);
```

```
        $self->recover($self->input, $exception);
        $exception = undef;
    }<\n>
<endif>
<endif>
<endif>
    <if(trace)>$self->traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <memoize()>
    <ruleScopeCleanUp()>
    <finally>
    if ($exception) {
        croak $exception;
        #$exception->rethrow();
    }
    <@postamble()>
    return <ruleReturnValue()>;
}
# $ANTLR end <ruleName>
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
    <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
my
 $retval = <returnType()>->new();
$retval->set_start($self->input->LT(1));<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
my $<a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
my $<ruleDescriptor.name>_start_index = $self->input->index();
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.pop();}; separator="\n">
```

```
<ruleDescriptor.ruleScope:{<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
   :{my $<it.label.text> = undef;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{List list_<it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it);
 separator="\n">
<ruleDescriptor.ruleListLabels:{ll|RuleReturnScope <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{my $<it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{List list_<it.label.text>=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
$<ruleDescriptor.singleValueReturnName>
<else>
$retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
$retval->set_stop($self->input->LT(-1));<\n>
<endif>
<endif>
>>
```

```
memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( backtracking>0 ) { memoize(input,
 <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start <ruleName>
sub m_<ruleName> {
# <ruleDescriptor.parameterScope:parameterScope(scope=it)>
   my ($self) = @_;
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleDeclarations()>
   eval {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       my $_type = <ruleName>;
       my $_channel = $self->DEFAULT_TOKEN_CHANNEL;
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       $self->state->type($_type);
       $self->state->channel($_channel);
       <(ruleDescriptor.actions.after):execAction()>
<endif>
    };
   <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
   <memoize()>

   if ($EVAL_ERROR) {
       croak $EVAL_ERROR;
   }
}
# $ANTLR end <ruleName>
>>
```

```
/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
sub m_tokens {
   my ($self) = @_;
   <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
my $alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch ($alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
my $alt<decisionNumber>
 = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch ($alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
```

```
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
# <fileName>:<description>
my $cnt<decisionNumber> = 0;
<decls>
<@preloop()>
LOOP<decisionNumber>:
while (1) {
   my $alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch ($alt<decisionNumber>) {
     <alts:altSwitchCase()>

  else {
        if ( $cnt<decisionNumber> >= 1 ) { last LOOP<decisionNumber> }
        <ruleBacktrackFailure()>
          my $eee =
              ANTLR::Runtime::EarlyExitException->new(<decisionNumber>, $self->input);
          <@earlyExitException()>
          croak $eee;
       }
   }
   ++$cnt<decisionNumber>;
}
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
<decls>
```

```
<@preloop()>
LOOP<decisionNumber>:
while (1) {
   my $alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch ($alt<decisionNumber>) {
     <alts:altSwitchCase()>
     else { last LOOP<decisionNumber> }
   }
}
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can
 just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> {
   <@prealt()>
   <it>
}<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
# <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
```

```
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)>$<label> =<endif>$self->match($self->input,
 <token>, $FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList();
list_<label>.add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
$self->match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
$self->match_range(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
```

```
<if(label)>
<if(LEXER)>
<label>= $self->input->LA(1);<\n>
<else>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<endif>
if ( <s> ) {
  $self->input->consume();
  <postmatchCode>
<if(!LEXER)>
   $self->state->error_recovery(0);
<endif>
  <if(backtracking)>failed=false;<endif>
}
else {
  <ruleBacktrackFailure()>
  my $mse =
     ANTLR::Runtime::MismatchedSetException->new(undef, $self->input);
  <@mismatchedSetException()>
<if(LEXER)>
  $self->recover($mse);
  $mse->throw();
<else>
  $mse->throw();
  <! use following code to make it recover inline; remove throw mse;
  $self->recoverFromMismatchedSet($self->input, $mse, $FOLLOW_set_in_<ruleName><elementIndex>);
  !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
int <label>Start = getCharIndex();
$self->match(<string>); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE,
Token.DEFAULT_CHANNEL, <label>Start, getCharIndex()-1);
<else>
$self->match(<string>); <checkRuleBacktrackFailure()><\n>
```

```
<endif>
>>

wildcard(label,elementIndex)
 ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = $self->input->LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
$self->push_follow($FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
$<label> = $self-><rule.name>(<args; separator=", ">);<\n>
<else>
$self-><rule.name>(<args; separator=", ">);<\n>
<endif>
$self->state->_fsp($self->state->_fsp - 1);
<checkRuleBacktrackFailure()>
>>

/**
 ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
```

```
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
$self->m_<rule>(<args; separator=", ">); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
$self->m_<rule.name>(<args; separator=", ">); <checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
match(EOF);
 <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>)
 ) {
   <ruleBacktrackFailure()>
   throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels">
else {
<if(eotPredictsAlt)>
   $alt<decisionNumber> = <eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   my $nvae =
       ANTLR::Runtime::NoViableAltException->new({
           grammar_decision_description => "<description>",
           decision_number => <decisionNumber>,
           state_number => <stateNumber>,
           input => $self->input,
       });<\n>
   <@noViableAltException()>
   croak $nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X
 ) match(X);" and that's it.
*/
```

```
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
my $LA<decisionNumber>_<stateNumber> = $self->input->LA(<k>);<\n>
<edges; separator="\nels"><\n>
<if(eotPredictsAlt)>
<if(!edges)>
$alt<decisionNumber> = <eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
   $alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted
 */
dfaAcceptState(alt) ::= "$alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k>) ) {
   <edges; separator="\n">
   else {
```

```
   <if(eotPredictsAlt)>
       $alt<decisionNumber> = <eotPredictsAlt>;
   <else>
       <ruleBacktrackFailure()>
       my $nvae =
           ANTLR::Runtime::NoViableAltException->new({
               grammar_decision_description => "<description>",

         decision_number => <decisionNumber>,
               state_number => <stateNumber>,
               input => $self->input,
           });<\n>
       <@noViableAltException()>
       croak $nvae;<\n>
   <endif>
   }
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k>) ) {
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( $self->input->LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
else { $alt<decisionNumber> = <eotPredictsAlt> }<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
case [<labels:{ <it> }; separator=", ">] { <targetState> }
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description)
 ::= <<
$alt<decisionNumber> = $self->dfa<decisionNumber>->predict($self->input);
>>
```

```
/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
Readonly my $DFA<dfa.decisionNumber>_eot => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedEOT; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_eof => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedEOF; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_min => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedMin; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_max => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedMax;
 separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_accept => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedAccept; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_special => ANTLR::Runtime::DFA->unpack_rle([
<dfa.javaCompressedSpecial; separator=", "> ]);
Readonly my $DFA<dfa.decisionNumber>_transition => [
<dfa.javaCompressedTransition:{s|ANTLR::Runtime::DFA->unpack_rle([ <s; separator=", "> ])}; separator=", "> ];

{
package <name>::DFA<dfa.decisionNumber>;
use ANTLR::Runtime::Class;

use strict;
use warnings;

extends 'ANTLR::Runtime::DFA';

sub BUILD {
   my $self = shift;
   my $param_ref = __PACKAGE__->unpack_params(@_, {
       spec => [
           {
              name => 'recognizer',
              isa  => 'ANTLR::Runtime::BaseRecognizer'
           },
       ]
   });

   $self->recognizer($param_ref->{recognizer});
   $self->decision_number(<dfa.decisionNumber>);
```

```
  $self->eot($DFA<dfa.decisionNumber>_eot);
  $self->eof($DFA<dfa.decisionNumber>_eof);
   $self->min($DFA<dfa.decisionNumber>_min);
  $self->max($DFA<dfa.decisionNumber>_max);
  $self->accept($DFA<dfa.decisionNumber>_accept);
  $self->special($DFA<dfa.decisionNumber>_special);
  $self->transition($DFA<dfa.decisionNumber>_transition);
}

sub get_description {
  return "<dfa.description>";
}

<@errorMethod()>

<if(dfa.specialStateSTs)>
sub special_state_transition {
  my ($self, $param_ref) = unpack_params(@_, {
    spec => [
      {
        name => 's',
        type => SCALAR,
      },
      {
        name => 'input',
        isa  => 'ANTLR::Runtime::IntStream',
      }
    ]
  });
  my $s = $param_ref->{s};
  my $input = $param_ref->{input};

  switch ($s) {
    <dfa.specialStateSTs:{state |
    case <i0> \{ <! compressed special state numbers 0..n-1 !>
      <state>}; separator="\n">
    }
  }

<if(backtracking)>
  if ($self->state->backtracking
 > 0) {
    $self->state->failed = 1;
    return -1;
  }<\n>
<endif>

  my $nvae =
        ANTLR::Runtime::NoViableAltException->new({
            grammar_decision_description => $self->get_description(),
            decision_number => <dfa.decisionNumber>,
            state_number => $s,
            input => $input,
        });<\n>
    $self->error($nvae);
    $nvae->throw();
    }<\n>
<endif>
}<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
my $input = $self->input;
my $LA<decisionNumber>_<stateNumber> = $input->LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
my $index<decisionNumber>_<stateNumber> = $input->index();
$input->rewind();<\n>
<endif>
s = -1;
<edges; separator="\nels">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if
 ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>
```

```
// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left> && <right>)"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o | ||<o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "$LA<decisionNumber>_<stateNumber> eq <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
 *  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "$self->input->LA(<k>) eq <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
($LA<decisionNumber>_<stateNumber> ge <lower> && $LA<decisionNumber>_<stateNumber> le <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "($self->input->LA(<k>) ge
<lower> && $self->input->LA(<k>) le <upper>)"

setTest(ranges) ::= "<ranges; separator=\" || \">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>
```

```
returnType() ::=
 <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.name>_return
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "undef".
*/
initValue(typeName) ::= <<
<if(typeName)>
<perlTypeInitMap.(typeName)>
<else>
undef
<endif>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
my $<label.label.text> = <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop
 tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
```

```
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
{
  package <returnType()>;
  use ANTLR::Runtime::Class;

  extends 'ANTLR::Runtime::<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope';

  <scope.attributes:{public <it.decl>;}; separator="\n">
  <@ruleReturnMembers()>
}
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{$<it.name>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "$<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "$<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
 =<expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name> =<expr>;
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
```

```
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
$<scope>.<attr.name>
<else>
$<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
$<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
 =<expr>;
<else>
$<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "$<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "$<scope>->get_text()"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
```

```
ruleLabelPropertyRef_start(scope,attr) ::= "((<labelType>)<scope>.start)"
ruleLabelPropertyRef_stop(scope,attr) ::= "((<labelType>)<scope>.stop)"
ruleLabelPropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)<scope>.tree)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
  input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
 input.getTreeAdaptor().getTokenStopIndex(<scope>.start))
<else>
substr($self->input, $<scope>->start, $<scope>->stop)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "$<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>.getText()"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType>)retval.stop)"
rulePropertyRef_tree(scope,attr)
 ::= "((<ASTLabelType>)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(retval.start),
 input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "$_type"
lexerRulePropertyRef_line(scope,attr) ::= "tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "$_channel"
lexerRulePropertyRef_start(scope,attr) ::= "tokenStartCharIndex"
```

```
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_self(scope,attr) ::= "$self"

// setting $st and $tree is allowed in local rule. everything
 else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"


/** How to execute an action */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
if ( backtracking==0 ) {
 <action>
}
<endif>
<else>
<action>
<endif>
>>

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
Readonly my $<name> => ANTLR::Runtime::BitSet->new({ words64 => [ <words64:{'<it>'};separator=", "> ]
});<\n>
>>

codeFileExtension() ::= ".pm"

true() ::= "1"
false() ::= "0"
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Perl5/Perl5.stg

No license file was found, but licenses were detected in source scan.

```
*
* Conversion to C#:
* Copyright (c) 2008-2009 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
 OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/
group ST;

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr3.ST;
using Antlr3.ST.Language;
>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
public StringTemplate
 st;
public object getTemplate() { return st; }
public override string ToString() { return (st==null) ? null : st.ToString(); }
>>

@genericParser.members() ::= <<
```

```
<@super.members()>
protected StringTemplateGroup templateLib = new StringTemplateGroup("<name>Templates",
typeof(AngleBracketTemplateLexer) );

public StringTemplateGroup TemplateLib
{
 get { return templateLib; }
 set { templateLib = value; }
}

///** allows convenient multi-value initialization:
// *  "new STAttrMap().put(...).put(...)"
// */
//public static class STAttrMap extends HashMap {
//  public STAttrMap put(String attrName, object value) {
//    super.put(attrName, value);
//    return this;
//  }
//  public STAttrMap put(String attrName, int value) {
//    super.put(attrName, new Integer(value));
//    return this;
//  }
//}
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".getTemplate()",...)>
>>

rewriteTemplate(alts)
 ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
 <alts:rewriteTemplateAlt(); separator="else ">
 <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
```

```
<if(TREE_PARSER)>
((TokenRewriteStream)input.TokenStream).replace(
 input.TreeAdaptor.GetTokenStartIndex(retval.start),
 input.TreeAdaptor.GetTokenStopIndex(retval.start),
 retval.st );
<else>
((TokenRewriteStream)input).replace(
 ((IToken)retval.start).TokenIndex,
 input.LT(-1).TokenIndex,
 retval.st );
<endif>
>>

rewriteTemplateAlt() ::= <<
// <it.description>
<if(it.pred)>
if (<it.pred>)
{
 retval.st = <it.alt>;
}<\n>
<else>
{
 retval.st = <it.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
null;
>>

/** Invoke a template with a set of attribute name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because
 the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args) ::= <<
templateLib.getInstanceOf("<name>"<if(args)>,
 new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
 <endif>)
>>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <<
templateLib.getInstanceOf(<expr><if(args)>,
 new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
 <endif>)
>>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <<
new StringTemplate(templateLib, "<template>"<if(args)>,
 new STAttrMap()<args:{a | .put("<a.name>", <a.value>)}>
 <endif>)
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr) ::= <<
(<st>).setAttribute("<attrName>",<expr>);
>>

/** Translate %{stringExpr}
 */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(templateLib,<stringExpr>)
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/ST.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/CompositeGrammarTree.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/CompositeGrammar.java
No license file was found, but licenses were detected in source scan.

```
*/
/** Template overrides to add debugging to normal Java output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
group Dbg;

@outputFile.debugPreprocessor() ::= "#define ANTLR_DEBUG"

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.Runtime.Debug;
using IOException = System.IO.IOException;
>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
public static readonly string[]
 ruleNames = new string[] {
   "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n    ", separator=", ">
};<\n>
<endif>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
   private int ruleLevel = 0;
   public int RuleLevel {
 get { return ruleLevel; }
   }
   public void IncRuleLevel() { ruleLevel++; }
   public void DecRuleLevel() { ruleLevel--; }
<if(profile)>
   <ctorForProfilingRootGrammar()>
<else>
   <ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else> <! imported grammar !>
   public int RuleLevel {
 get { return <grammar.delegators:{g| <g:delegateName()>}>.RuleLevel; }
   }
   public void IncRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.IncRuleLevel(); }
   public void DecRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.DecRuleLevel(); }
   <ctorForDelegateGrammar()>
<endif>
<if(profile)>
override public bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
   ((Profiler)dbg).ExamineRuleMemoization(input,
 ruleIndex, <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
   return base.AlreadyParsedRule(input, ruleIndex);
```

```
}<\n>
override public void Memoize(IIntStream input,
                int ruleIndex,
                int ruleStartIndex)
{
  ((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
  base.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected bool EvalPredicate(bool result, string predicate)
{
  dbg.SemanticPredicate(result, predicate);
  return result;
}<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
public <name>(<inputStreamType> input)
  : this(input, DebugEventSocketProxy.DEFAULT_DEBUGGER_PORT, new RecognizerSharedState()) {
}

public <name>(<inputStreamType> input, int port, RecognizerSharedState state)

 : base(input, state) {
  <createListenerAndHandshake()>
  <parserCtorBody()>
  <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
  <@finally()>
}<\n>
>>

@parserCtorBody.initializeCyclicDFAs() ::= <<
InitializeCyclicDFAs(dbg);
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
public <name>(<inputStreamType> input) {
  this(input, new Profiler(null), new RecognizerSharedState());
}

public <name>(<inputStreamType> input, IDebugEventListener dbg, RecognizerSharedState state)
  : base(input, dbg, state) {
  Profiler p = (Profiler)dbg;
```

```
   p.setParser(this);
   <parserCtorBody()>
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state, this<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
   <@finally()>
}
<\n>
>>


/** Basically we don't want to
 set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
public <name>(<inputStreamType> input, IDebugEventListener dbg, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
   : base(input, dbg, state) {
   <parserCtorBody()>
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
}<\n>
>>

ctorForPredefinedListener() ::= <<
public <name>(<inputStreamType> input, IDebugEventListener dbg)
   : <@superClassRef>base(input, dbg, new RecognizerSharedState())<@end> {
<if(profile)>
   Profiler p = (Profiler)dbg;
   p.setParser(this);
<endif>
   <parserCtorBody()>
   <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
   <@finally()>
}<\n>
>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy
 dbg = new DebugEventSocketProxy(this, port, input.TreeAdaptor);
<else>
DebugEventSocketProxy dbg = new DebugEventSocketProxy(this, port, null);
<endif>
DebugListener = dbg;
try
{
   dbg.Handshake();
}
catch (IOException ioe)
{
   ReportError(ioe);
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

@rule.preamble() ::= <<
try {
 dbg.EnterRule(GrammarFileName, "<ruleName>");
 if ( RuleLevel==0 ) {dbg.Commence();}
 IncRuleLevel();
 dbg.Location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>);<\n>
>>

@lexer.debugInitializeCyclicDFAs() ::= "IDebugEventListener dbg"

@lexer.debugAddition() ::= ", dbg"

@genericParser.debugInitializeCyclicDFAs() ::= "IDebugEventListener dbg"

@genericParser.debugAddition() ::= ", dbg"

@rule.postamble() ::= <<
dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>);<\n>
}
finally {
   dbg.ExitRule(GrammarFileName, "<ruleName>");
   DecRuleLevel();
   if ( RuleLevel==0 ) {dbg.Terminate();}
}<\n>
>>

@synpred.start()
 ::= "dbg.BeginBacktrack(state.backtracking);"

@synpred.stop() ::= "dbg.EndBacktrack(state.backtracking, success);"

// Common debug event triggers used by region overrides below

enterSubRule() ::=
   "try { dbg.EnterSubRule(<decisionNumber>);<\n>"

exitSubRule() ::=
   "} finally { dbg.ExitSubRule(<decisionNumber>); }<\n>"
```

```
enterDecision() ::=
   "try { dbg.EnterDecision(<decisionNumber>);<\n>"

exitDecision() ::=
   "} finally { dbg.ExitDecision(<decisionNumber>); }<\n>"

enterAlt(n) ::= "dbg.EnterAlt(<n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop()
 ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
   "dbg.RecognitionException(eee<decisionNumber>);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
  "dbg.Location(<it.line>,<it.pos>);"

@matchSet.mismatchedSetException() ::=
  "dbg.RecognitionException(mse);"

@dfaState.noViableAltException() ::= "dbg.RecognitionException(nvae_d<decisionNumber>s<stateNumber>);"

@dfaStateSwitch.noViableAltException() ::=
"dbg.RecognitionException(nvae_d<decisionNumber>s<stateNumber>);"

dfaDecision(decisionNumber,description) ::= <<
try
{
   isCyclicDecision = true;
  <super.dfaDecision(...)>
}
catch (NoViableAltException nvae)
{
  dbg.RecognitionException(nvae);
  throw nvae;
}
>>

@cyclicDFA.dbgCtor() ::= <<
  public DFA<dfa.decisionNumber>(BaseRecognizer recognizer, IDebugEventListener dbg) : this(recognizer)
  {
 this.dbg = dbg;
  }
>>

@cyclicDFA.debugMember() ::= <<
IDebugEventListener dbg;

>>

@cyclicDFA.errorMethod() ::= <<
public override void Error(NoViableAltException nvae)
{
  dbg.RecognitionException(nvae);
}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
EvalPredicate(<pred>,"<description>")
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/Dbg.stg
No license file was found, but licenses were detected in source scan.

```
/*
New style messages. This file contains the actual layout of the messages emitted by ANTLR.
The text itself is coming out of the languages/*stg files, according to the chosen locale.
This file contains the format that mimicks GCC output.
*/

group gnu;

location(file, line, column) ::= "<file>:<line>:"

message(id, text) ::= "<text> (<id>)"

report(location, message, type) ::= "<location> <type>: <message>"

wantsSingleLineMessage() ::= "true"
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/templates/messages/formats/gnu.stg
No license file was found, but licenses were detected in source scan.

```
/*
[The "BSD licence"]
Copyright (c) 2007 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/
group ASTParser;

// TOKEN AST STUFF

/** ID and output=AST */
```

```
tokenRef(token,label,hetero,elementIndex)
 ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( backtracking == 0 ) {<endif>
_<label>_tree = (<ASTLabelType>)[treeAdaptor newTreeWithToken:_<label>];
[treeAdaptor addChild:_<label>_tree toTree:root_0];
[_<label>_tree release];
<if(backtracking)>}<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,hetero,elementIndex) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( backtracking == 0 ) {<endif>
_<label>_tree = (<ASTLabelType>)[treeAdaptor newTreeWithToken:_<label>];
root_0 = (<ASTLabelType>)[treeAdaptor makeNode:_<label>_tree parentOf:root_0];
[_<label>_tree release];
<if(backtracking)>}<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,hetero,elementIndex) ::=
 <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,hetero,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

// TODO: add support for heterogeneous trees

matchSet(s,label,hetero,elementIndex,postmatchCode)
 ::= <<
<super.matchSet(..., postmatchCode={
<if(backtracking)>if (backtracking == 0) {<endif>
_<label>_tree = (<ASTLabelType>)[treeAdaptor newTreeWithToken:_<label>];
[treeAdaptor addChild:_<label>_tree toTree:root_0];
[_<label>_tree release];
<if(backtracking)>}<endif>
})>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(backtracking)>if (backtracking == 0) {<endif>
_<label>_tree = (<ASTLabelType>)[treeAdaptor newTreeWithToken:_<label>];
root_0 = (<ASTLabelType>)[treeAdaptor makeNode:_<label>_tree parentOf:root_0];
[_<label>_tree release];
<if(backtracking)>}<endif>
})>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (backtracking == 0) <endif>[treeAdaptor
 addChild:[_<label> tree] toTree:root_0];
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
```

```
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (backtracking == 0) <endif>root_0 = (<ASTLabelType>)[treeAdaptor makeNode:[_<label>
tree] parentOf:root_0];
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem="["+label+" tree]",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem="["+label+" tree]",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem="["+label+" tree]",...)>
>>

// WILDCARD AST

wildcard(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if
 (backtracking == 0) {<endif>
_<label>_tree = (<ASTLabelType>)[treeAdaptor newTreeWithToken:_<label>];
[treeAdaptor addChild:_<label>_tree toTree:root_0];
[_<label>_tree release];
<if(backtracking)>}<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if (backtracking == 0) {<endif>
_<label>_tree = (<ASTLabelType>)[treeAdaptor newTreeWithToken:_<label>];
root_0 = (<ASTLabelType>)[treeAdaptor makeNode:_<label>_tree parentOf:root_0];
[_<label>_tree release];
<if(backtracking)>}<endif>
>>
```

```
createNodeFromToken(label,hetero) ::= <<
<if(hetero)>
new <hetero>(<label>) <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType>)adaptor.create(<label>)
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ObjC/ASTParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2008 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/ANTLRParser.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/ANTLRTokenTypes.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/ErrorManager.java

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NFAConversionThread.java
*
/opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarNonDeterminismMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/ANTLRTreePrinterTokenTypes.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/ANTLRLexer.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/AnalysisTimeoutException.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/PredicateLabel.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/MutableInteger.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/TreeToNFAConverter.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/AssignTokenTypesWalkerTokenTypes.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarAnalysisAbortedMessage.java
*
/opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/Interp.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarAST.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarSyntaxMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/DOTGenerator.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/Attribute.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/DefineGrammarItemsWalker.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/NonRegularDecisionMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/Rule.java
*
/opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/AssignTokenTypesWalker.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/TreeToNFAConverterTokenTypes.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/CodeGenTreeWalker.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/AssignTokenTypesBehavior.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/Message.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/CodeGenTreeWalkerTokenTypes.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarSanity.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/Grammar.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/DefineGrammarItemsWalkerTokenTypes.java

*
/opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/Barrier.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/ANTLRErrorListener.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarSemanticsMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/Interpreter.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/ToolMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/LeftRecursionCyclesMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/NameSpaceChecker.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/MultiMap.java
*
/opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarDanglingStateMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarReport.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarInsufficientPredicatesMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarUnreachableAltsMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/RandomPhrase.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NonLLStarDecisionException.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/LL1Analyzer.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/RuleLabelScope.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/FASerializer.java
*
/opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/LL1DFA.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/AnalysisRecursionOverflowException.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/RecursionOverflowMessage.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/AttributeScope.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/grammar/v2/ANTLRTreePrinter.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/ActionLabel.java
No license file was found, but licenses were detected in source scan.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
group CSharp2 implements ANTLRCore;

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,

   fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@debugPreprocessor()>
// The variable 'variable' is assigned but its value is never used.
#pragma warning disable 168, 219
// Unreachable code detected.
#pragma warning disable 162

<actions.(actionScope).header>
```

```
<@imports>
using System;
using Antlr.Runtime;
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
<endif>
using IList   = System.Collections.IList;
using ArrayList  = System.Collections.ArrayList;
using Stack   = Antlr.Runtime.Collections.StackList;

<if(backtracking)>
using IDictionary = System.Collections.IDictionary;
using Hashtable  = System.Collections.Hashtable;
<endif>

<@end>

<if(actions.(actionScope).namespace)>
namespace <actions.(actionScope).namespace>
{
<endif>

<docComment>
<recognizer>
<if(actions.(actionScope).namespace)>
}
<endif>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="IToken",

 filterMode, superClass="Lexer") ::= <<
public partial class <grammar.recognizerName> : <@superClassName><superClass><@end> {
  <tokens:{public const int <it.name> = <it.type>;}; separator="\n">
  <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
  <actions.lexer.members>

  // delegates
  <grammar.delegates:
      {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  // delegators
  <grammar.delegators:
      {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

  public <grammar.recognizerName>() <! needed by subclasses !>
  {
```

```
  InitializeCyclicDFAs();
   }
   public <grammar.recognizerName>(ICharStream input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>)
  : this(input, null<grammar.delegators:{g|, <g:delegateName()>}>) {
   }
   public <grammar.recognizerName>(ICharStream input, RecognizerSharedState state<grammar.delegators:{g|,
 <g.recognizerName> <g:delegateName()>}>)
  : base(input, state) {
  InitializeCyclicDFAs(); <! Necessary in C#??? Not removed yet. !>
<if(memoize)>
<if(grammar.grammarIsRoot)>
      state.ruleMemo = new Hashtable[<numRules>+1];<\n> <! index from 1..n !>
<endif>
<endif>
      <grammar.directDelegates:
       {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
      <grammar.delegators:
       {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
      <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
   }

   override public string GrammarFileName
   {
    get { return "<fileName>";}
   }

<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>

   <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">
 private void InitializeCyclicDFAs(<@debugInitializeCyclicDFAs()>)
 {

    <cyclicDFAs:{dfa | this.dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this<@debugAddition()>);}; separator="\n">
     <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>this.dfa<dfa.decisionNumber>.specialStateTransitionHandler = new
DFA.SpecialStateTransitionHandler(DFA<dfa.decisionNumber>_SpecialStateTransition);<endif>};
separator="\n">
 }

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
```

```
}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
override public IToken NextToken()
{
   while (true)
 {
       if ( input.LA(1) == (int)CharStreamConstants.EOF )
  {
           return Token.EOF_TOKEN;
       }

     state.token
 = null;
  state.channel = Token.DEFAULT_CHANNEL;
       state.tokenStartCharIndex = input.Index();
       state.tokenStartCharPositionInLine = input.CharPositionInLine;
       state.tokenStartLine = input.Line;
     state.text = null;
       try
  {
           int m = input.Mark();
           state.backtracking = 1; <! means we won't throw slow exception !>
           state.failed = false;
           mTokens();
           state.backtracking = 0;
<!
   mTokens backtracks with synpred at backtracking==2
           and we set the synpredgate to allow actions at level 1.
!>
           if ( state.failed )
   {
             input.Rewind(m);
               input.Consume(); <! // advance one char and try again !>
           }
           else
   {
    Emit();
               return state.token;
           }
```

```
        }
        catch (RecognitionException re)
    {
            // shouldn't happen in backtracking mode, but...
            ReportError(re);
            Recover(re);
         }
    }
}

override public void Memoize(IIntStream input, int ruleIndex, int ruleStartIndex)
{
 if ( state.backtracking > 1 )
  base.Memoize(input, ruleIndex, ruleStartIndex);
}

override public bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
 if ( state.backtracking>1 )
  return base.AlreadyParsedRule(input, ruleIndex);
 return false;
}
>>

actionGate() ::= "(state.backtracking==0)"

filteringActionGate() ::= "(state.backtracking == 1)"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
             bitsets, inputStreamType, superClass, filterMode,
             ASTLabelType="object", labelType, members, rewriteElementType) ::= <<
public partial class <grammar.recognizerName> : <@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
   public static readonly string[] tokenNames = new string[]
 {
       "\<invalid>",
  "\<EOR>",
  "\<DOWN>",
  "\<UP>",
  <tokenNames; separator=", \n">
   };<\n>
<endif>

    <tokens:{public const int <it.name> = <it.type>;}; separator="\n">

   // delegates
  <grammar.delegates:
       {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  // delegators
  <grammar.delegators:
       {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

  <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
  <@members>
  <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>

  public <grammar.recognizerName>(<inputStreamType> input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>)
 : this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>) {
  }

  public <grammar.recognizerName>(<inputStreamType> input, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
 : base(input, state) {
     <parserCtorBody()>
     <grammar.directDelegates:
      {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
     <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
     <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
  }
  <@end>

  override public string[] TokenNames {
 get { return <grammar.composite.rootGrammar.recognizerName>.tokenNames; }
  }

  override public string GrammarFileName {
 get { return "<fileName>"; }
  }

  <members>

  <rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
 appear to be defined in this recognizer. !>
  // Delegated rules
  <grammar.delegatedRules:{ruleDescriptor|
  public <returnType()> <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) //
throws RecognitionException
  \{
    <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>}; separator=", ">);
   \}}; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">
 private void InitializeCyclicDFAs(<@debugInitializeCyclicDFAs()>)
 {
    <cyclicDFAs:{dfa | this.dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this<@debugAddition()>);}; separator="\n">
     <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>this.dfa<dfa.decisionNumber>.specialStateTransitionHandler = new
DFA.SpecialStateTransitionHandler(DFA<dfa.decisionNumber>_SpecialStateTransition);<endif>};
separator="\n">
 }

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                 words64=it.bits)>
}
>>

parserCtorBody() ::= <<
<@initializeCyclicDFAs>InitializeCyclicDFAs();<@end>
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo
 = new Hashtable[<length(grammar.allImportedRules)>+1];<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType, superClass="Parser",
labelType="IToken", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
```

```
>>

/** A simpler version of a rule template that is specific to the imaginary
*
 rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start "<ruleName>"
public void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) {
   <ruleLabelDefs()>
<if(trace)>
   TraceIn("<ruleName>_fragment", <ruleDescriptor.index>);
   try
   {
       <block>
   }
   finally
   {
       TraceOut("<ruleName>_fragment", <ruleDescriptor.index>);
   }
<else>
   <block>
<endif>
}
// $ANTLR end "<ruleName>"
>>

synpredDecls(name) ::= <<
SynPredPointer <name>;<\n>
>>

synpred(name) ::= <<
public bool <name>()
{
   state.backtracking++;
   <@start()>
   int start = input.Mark();
   try
   {
       <name>_fragment();
 // can never throw exception
   }
   catch (RecognitionException re)
```

```
  {
    Console.Error.WriteLine("impossible: "+re);
  }
  bool success = !state.failed;
  input.Rewind(start);
  <@stop()>
  state.backtracking--;
  state.failed = false;
  return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( (state.backtracking > 0) && AlreadyParsedRule(input, <ruleDescriptor.index>) )
{
 return <ruleReturnValue()>;
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if ( state.backtracking > 0 ) {state.failed = true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for
 multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
public <returnType()> <ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) // throws RecognitionException [1]
{
  <if(trace)>TraceIn("<ruleName>", <ruleDescriptor.index>);<endif>
  <ruleScopeSetUp()>
  <ruleDeclarations()>
  <ruleLabelDefs()>
  <ruleDescriptor.actions.init>
  <@preamble()>
  try
 {
    <ruleMemoization(name=ruleName)>
    <block>
    <ruleCleanUp()>
    <(ruleDescriptor.actions.after):execAction()>
  }
<if(exceptions)>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
  <actions.(actionScope).rulecatch>
<else>
  catch (RecognitionException re)
 {
    ReportError(re);
     Recover(input,re);
 <@setErrorReturnValue()>
  }<\n>
<endif>
<endif>
<endif>
  finally
 {
    <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <memoize()>
    <ruleScopeCleanUp()>
    <finally>
  }
  <@postamble()>
  return <ruleReturnValue()>;
}
// $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
  <e.action>
}
>>
```

```
ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = new <returnType()>();
retval.Start = input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.Index();
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.Push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.Push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::=
 <<
<ruleDescriptor.useScopes:{<it>_stack.Pop();}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.Pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{<labelType> <it.label.text> = null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{IList list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll:ruleLabelDef(label=it)> <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<labelType> <it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
```

```
    :{IList list_<it.label.text> = null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.Stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking > 0 )
{
 Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex);
}
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
public void m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) // throws
RecognitionException [2]
{
 <ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
 <if(trace)>TraceIn("<ruleName>",
 <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
```

```
  try
  {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       int _type = <ruleName>;
 int _channel = DEFAULT_TOKEN_CHANNEL;
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       state.type = _type;
       state.channel = _channel;
       <(ruleDescriptor.actions.after):execAction()>
<endif>
   }
   finally
 {
       <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
       <ruleScopeCleanUp()>
       <memoize()>
   }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
override public void mTokens() // throws RecognitionException

{
   <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber> = <maxAlt>;
<decls>
```

```
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
{
  <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>)
{
  <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single
 alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int cnt<decisionNumber> = 0;
```

```
<decls>
<@preloop()>
do
{
   int alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
 {
  <alts:altSwitchCase()>
  default:
      if ( cnt<decisionNumber> >= 1 ) goto loop<decisionNumber>;
      <ruleBacktrackFailure()>
            EarlyExitException eee<decisionNumber> =
                new EarlyExitException(<decisionNumber>, input);
            <@earlyExitException()>
            throw eee<decisionNumber>;
   }
   cnt<decisionNumber>++;
} while (true);

loop<decisionNumber>:
 ; // Stops C# compiler
 whining that label 'loop<decisionNumber>' has no statements
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
do
{
   int alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
 {
  <alts:altSwitchCase()>
  default:
      goto loop<decisionNumber>;
   }
} while (true);

loop<decisionNumber>:
 ; // Stops C# compiler whining that label 'loop<decisionNumber>' has no statements
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
 *  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> :
   <@prealt()>
   <it>
   break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
 <@declarations()>
 <elements:element()>
 <rew>
 <@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
```

```
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label>=(<labelType>)<endif>Match(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex
>); <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == null) list_<label> = new ArrayList();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label> = (<labelType>)input.LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
  input.Consume();
  <postmatchCode>
<if(!LEXER)>
  state.errorRecovery = false;
<endif>
  <if(backtracking)>state.failed = false;<endif>
}
else
{
  <ruleBacktrackFailure()>
  MismatchedSetException mse = new MismatchedSetException(null,input);
   <@mismatchedSetException()>
<if(LEXER)>
  Recover(mse);
  throw mse;
<else>
  throw mse;
  <! use following code to make it recover inline; remove throw mse;
  RecoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
  !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
int <label>Start = CharIndex;
Match(<string>); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start, CharIndex-1);
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label> = (<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
```

```
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/**
 Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
<label> = <if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
<else>
<if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
<endif>
state.followingStackPointer--;
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/**
 A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = CharIndex;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, CharIndex-1);
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = CharIndex;
Match(EOF);
 <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, CharIndex-1);
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1) == Token.DOWN )
{
   Match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
```

```
Match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) )
{
   <ruleBacktrackFailure()>
    throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
{
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   NoViableAltException nvae_d<decisionNumber>s<stateNumber> =
       new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
   <@noViableAltException()>
   throw nvae_d<decisionNumber>s<stateNumber>;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
```

```
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else
{
   alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression
 is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>) <endif>)
{
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
<edges; separator="\n">
 default:
<if(eotPredictsAlt)>
    alt<decisionNumber> = <eotPredictsAlt>;
    break;
<else>
    <ruleBacktrackFailure()>
    NoViableAltException nvae_d<decisionNumber>s<stateNumber> =
        new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
    <@noViableAltException()>
    throw nvae_d<decisionNumber>s<stateNumber>;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
switch ( input.LA(<k>) )
{
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
 default:
    alt<decisionNumber> = <eotPredictsAlt>;
    break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n">
 {
   <targetState>
   }
   break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
>>
```

```
/* Dump DFA tables.
*/
cyclicDFA(dfa) ::= <<
const string DFA<dfa.decisionNumber>_eotS =
   "<dfa.javaCompressedEOT; wrap="\"+\n    \"">";
const string
 DFA<dfa.decisionNumber>_eofS =
   "<dfa.javaCompressedEOF; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_minS =
   "<dfa.javaCompressedMin; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_maxS =
   "<dfa.javaCompressedMax; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_acceptS =
   "<dfa.javaCompressedAccept; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_specialS =
   "<dfa.javaCompressedSpecial; wrap="\"+\n    \"">}>";
static readonly string[] DFA<dfa.decisionNumber>_transitionS = {
       <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
};

static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
static readonly short[] DFA<dfa.decisionNumber>_eof =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
static readonly char[] DFA<dfa.decisionNumber>_max
 = DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
static readonly short[][] DFA<dfa.decisionNumber>_transition =
DFA.UnpackEncodedStringArray(DFA<dfa.decisionNumber>_transitionS);

protected class DFA<dfa.decisionNumber> : DFA
{
   <@debugMember()>
   public DFA<dfa.decisionNumber>(BaseRecognizer recognizer)
   {
       this.recognizer = recognizer;
       this.decisionNumber = <dfa.decisionNumber>;
       this.eot = DFA<dfa.decisionNumber>_eot;
       this.eof = DFA<dfa.decisionNumber>_eof;
       this.min = DFA<dfa.decisionNumber>_min;
       this.max = DFA<dfa.decisionNumber>_max;
       this.accept = DFA<dfa.decisionNumber>_accept;
       this.special = DFA<dfa.decisionNumber>_special;
```

```
        this.transition
 = DFA<dfa.decisionNumber>_transition;

    }
    <@dbgCtor()>

    override public string Description
    {
        get { return "<dfa.description>"; }
    }

    <@errorMethod()>
}<\n>
<if(dfa.specialStateSTs)>

protected internal int DFA<dfa.decisionNumber>_SpecialStateTransition(DFA dfa, int s, IIntStream _input)
//throws NoViableAltException
{
        <if(LEXER)>
        IIntStream input = _input;
        <endif>
        <if(PARSER)>
        ITokenStream input = (ITokenStream)_input;
        <endif>
        <if(TREE_PARSER)>
        ITreeNodeStream input = (ITreeNodeStream)_input;
        <endif>
 int _s = s;
    switch ( s )
    {
    <dfa.specialStateSTs:{state |
        case <i0> : <! compressed special state numbers 0..n-1 !>
            <state>}; separator="\n">
    }
<if(backtracking)>
    if (state.backtracking > 0) {state.failed = true; return -1;}<\n>
<endif>
    NoViableAltException nvae<dfa.decisionNumber> =
        new NoViableAltException(dfa.Description,
 <dfa.decisionNumber>, _s, input);
    dfa.Error(nvae<dfa.decisionNumber>);
    throw nvae<dfa.decisionNumber>;
}<\n>
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
```

```
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
int index<decisionNumber>_<stateNumber> = input.Index();
input.Rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.Seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s >= 0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>) <endif>) { s = <targetStateNumber>;
 }<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left> && <right>)"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o | || <o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
```

```
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber> \>= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt)
 ::= "(input.LA(<k>) \>= <lower> && input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= "<ranges; separator=\" || \">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected class <scope.name>_scope
{
   <scope.attributes:{protected internal <it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected class <scope.name>_scope
{
   <scope.attributes:{protected internal <it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>
```

```
/** Generate the C#
 type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>.<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
default(<typeName>)
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope)
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public class <ruleDescriptor:returnStructName()> :
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope
{
   <scope.attributes:{public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
};
<endif>
>>
```

```
parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack[<scope>_stack.Count-<negIndex>-1]).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack[<index>]).<attr.name>
<else>
((<scope>_scope)<scope>_stack.Peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack[<scope>_stack.Count-<negIndex>-1]).<attr.name> = <expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack[<index>]).<attr.name> = <expr>;
<else>
((<scope>_scope)<scope>_stack.Peek()).<attr.name>
 = <expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
((<scope> != null) ? <scope>.<attr.name> : <initValue(attr.type)>)
<else>
<scope>
<endif>
>>
```

```
returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> = <expr>;
<else>
<attr.name> = <expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/**
 ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "((<scope> != null) ? <scope>.Text : null)"
tokenLabelPropertyRef_type(scope,attr) ::= "((<scope> != null) ? <scope>.Type : 0)"
tokenLabelPropertyRef_line(scope,attr) ::= "((<scope> != null) ? <scope>.Line : 0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "((<scope> != null) ? <scope>.CharPositionInLine : 0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "((<scope> != null) ? <scope>.Channel : 0)"
tokenLabelPropertyRef_index(scope,attr) ::= "((<scope> != null) ? <scope>.TokenIndex : 0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int.Parse(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "((<scope> != null) ? ((<labelType>)<scope>.Start) : null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "((<scope> != null) ? ((<labelType>)<scope>.Stop) : null)"
ruleLabelPropertyRef_tree(scope,attr)
 ::= "((<scope> != null) ? ((<ASTLabelType>)<scope>.Tree) : null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
((<scope> != null) ? input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(<scope>.Start),
 input.TreeAdaptor.GetTokenStopIndex(<scope>.Start)) : null)
<else>
((<scope> != null) ? input.ToString((IToken)(<scope>.Start),(IToken)(<scope>.Stop)) : null)
<endif>
>>
```

```
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> != null) ? <scope>.ST : null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> != null) ? <scope>.Type : 0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> != null) ? <scope>.Line : 0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> != null) ? <scope>.CharPositionInLine : -1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "((<scope> != null) ? <scope>.Channel : 0)"
lexerRuleLabelPropertyRef_index(scope,attr)
 ::= "((<scope> != null) ? <scope>.TokenIndex : 0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> != null) ? <scope>.Text : null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int.Parse(<scope>.Text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.Start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType>)retval.Stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.Tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start) )
<else>
input.ToString((IToken)retval.Start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.ST"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr)
 ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.Tree = <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.ST = <expr>;"


/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
```

```
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
 <action>
}
<else>
<action>
<endif>
>>


/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::=
 <<
public static readonly BitSet <name> = new BitSet(new ulong[]{<words64:{<it>UL};separator=",">});<\n>
>>

codeFileExtension() ::= ".cs"

true() ::= "true"
false() ::= "false"
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/CSharp2.stg

No license file was found, but licenses were detected in source scan.

```
*/
group ActionScript implements ANTLRCore;

asTypeInitMap ::= [
 "int":"0",
 "uint":"0",
 "Number":"0.0",
 "Boolean":"false",
 default:"null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
     bitsets, buildTemplate, buildAST,
 rewriteMode, profile,
     backtracking, synpreds, memoize, numRules,
     fileName, ANTLRVersion, generatedTimestamp, trace,
     scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
package<if(actions.(actionScope).package)> <actions.(actionScope).package><endif> {
    <actions.(actionScope).header>
    <@imports>
import org.antlr.runtime.*;
<if(TREE_PARSER)>
    import org.antlr.runtime.tree.*;
<endif>
    <@end>

    <docComment>
    <recognizer>
}
>>
```

```
lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="Lexer") ::= <<
public class <grammar.recognizerName> extends
<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else><@superClassName><superClass
><@end><endif> {
   <tokens:{public static const <it.name>:int=<it.type>;}; separator="\n">
   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
   <actions.lexer.members>

   // delegates
   <grammar.delegates:
         {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   <last(grammar.delegators):{g|public var gParent:<g.recognizerName>;}>

   public function <grammar.recognizerName>(<grammar.delegators:{g|<g:delegateName()>:<g.recognizerName>,
}>input:CharStream = null, state:RecognizerSharedState = null) {
       super(input, state);
       <cyclicDFAs:cyclicDFACtor()>
<if(memoize)>
<if(grammar.grammarIsRoot)>
       this.state.ruleMemo = new Array(<numRules>+1);<\n> <! index from 1..n !>
<endif>
<endif>
       <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>this,
input, this.state);}; separator="\n">
       <grammar.delegators:
        {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
       <last(grammar.delegators):{g|gParent
 = <g:delegateName()>;}>
   }
   public override function get grammarFileName():String { return "<fileName>"; }

<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

}
>>
```

```
/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public override function nextToken():Token {
   while (true) {
       if ( input.LA(1)==CharStreamConstants.EOF ) {
           return TokenConstants.EOF_TOKEN;
       }
       this.state.token = null;
     this.state.channel = TokenConstants.DEFAULT_CHANNEL;
       this.state.tokenStartCharIndex = input.index;
       this.state.tokenStartCharPositionInLine
 = input.charPositionInLine;
       this.state.tokenStartLine = input.line;
     this.state.text = null;
       try {
           var m:int = input.mark();
           this.state.backtracking=1; <! means we won't throw slow exception !>
           this.state.failed=false;
           mTokens();
           this.state.backtracking=0;
           <! mTokens backtracks with synpred at backtracking==2
              and we set the synpredgate to allow actions at level 1. !>
           if ( this.state.failed ) {
               input.rewindTo(m);
               input.consume(); <! advance one char and try again !>
           }
           else {
               emit();
               return this.state.token;
           }
       }
       catch (re:RecognitionException) {
           // shouldn't happen in backtracking mode, but...
           reportError(re);
           recover(re);
       }
   }
   // Not reached - For ActionScript compiler
   throw new
 Error();
}

public override function memoize(input:IntStream,
  ruleIndex:int,
  ruleStartIndex:int):void
{
if ( this.state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
}

public override function alreadyParsedRule(input:IntStream, ruleIndex:int):Boolean {
if ( this.state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>

actionGate() ::= "this.state.backtracking==0"

filteringActionGate() ::= "this.state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass, filterMode,
            ASTLabelType="Object", labelType, members, rewriteElementType) ::= <<
public class <grammar.recognizerName> extends
<if(actions.(actionScope).superClass)><actions.(actionScope).superClass><else><@superClassName><superClass
><@end><endif> {
<if(grammar.grammarIsRoot)>
   public static const tokenNames:Array = [
       "\<invalid>",
 "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
   ];<\n>
<endif>
   <tokens:{public static const <it.name>:int=<it.type>;}; separator="\n">

   // delegates
   <grammar.delegates:
        {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|public var <g:delegateName()>:<g.recognizerName>;}; separator="\n">
   <last(grammar.delegators):{g|public var gParent:<g.recognizerName>;}>

   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
   <@members>
  <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
   public function <grammar.recognizerName>(<grammar.delegators:{g|<g:delegateName()>:<g.recognizerName>,
}>input:<inputStreamType>, state:RecognizerSharedState = null) {
       super(input, state);
       <cyclicDFAs:cyclicDFACtor()>
       <parserCtorBody()>
       <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>,
 }>this, input, this.state);}; separator="\n">
       <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
       <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
   }
   <@end>

   public override function get tokenNames():Array { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; }
   public override function get grammarFileName():String { return "<fileName>"; }

   <members>

   <rules; separator="\n\n">

   <! generate rule/method definitions for imported rules so they
      appear to be defined in this recognizer. !>
      // Delegated rules
   <grammar.delegatedRules:{ruleDescriptor|
       public function
<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):<returnType()> \{
<if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>};
 separator=", ">); \}}; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                 words64=it.bits)>
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new Array(<length(grammar.allImportedRules)>+1);<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType="Object",
superClass="Parser", labelType="Token", members={<actions.parser.members>}) ::= <<
```

```
<genericParser(inputStreamType="TokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules,
 bitsets, labelType={<ASTLabelType>}, ASTLabelType="Object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="TreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public final function <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):void {
   <ruleLabelDefs()>
<if(trace)>
   traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
   try {
       <block>
   }
   finally {

    traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
   }
<else>
   <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public final function <name>():Boolean {
   this.state.backtracking++;
   <@start()>
   var start:int = input.mark();
   try {
       <name>_fragment(); // can never throw exception
   } catch (re:RecognitionException) {
        trace("impossible: "+re);
    }
    var success:Boolean = !this.state.failed;
    input.rewindTo(start);
    <@stop()>
    this.state.backtracking--;
    this.state.failed=false;
    return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( this.state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>;
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (this.state.failed) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed,
 exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (this.state.backtracking>0) {this.state.failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
// $ANTLR start <ruleName>
// <fileName>:<description>
public final function <ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):<returnType()> {
    <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
    <ruleLabelDefs()>
    <ruleDescriptor.actions.init>
    <@preamble()>
    try {
```

```
    <ruleMemoization(name=ruleName)>
    <block>
    <ruleCleanUp()>
    <(ruleDescriptor.actions.after):execAction()>
  }
<if(exceptions)>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
  catch (re:RecognitionException) {
    reportError(re);
    recoverStream(input,re);
    <@setErrorReturnValue()>
  }<\n>
<endif>
<endif>
<endif>
  finally {
    <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <memoize()>
    <ruleScopeCleanUp()>
    <finally>
  }
  <@postamble()>
  return <ruleReturnValue()>;
}
// $ANTLR end <ruleName>
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
  <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
var retval:<returnType()> = new <returnType()>();
retval.start = input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
var <a.name>:<a.type> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
```

```
var <ruleDescriptor.name>_StartIndex:int = input.index;
<endif>
>>

ruleScopeSetUp()
 ::= <<
<ruleDescriptor.useScopes:{<it>_stack.push(new Object());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.push(new Object());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{var <it.label.text>:<labelType>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{var list_<it.label.text>:Array=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|var <ll.label.text>:RuleReturnScope = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
  ruleDescriptor.ruleLabels]
   :{var <it.label.text>:<labelType>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{var <it.label.text>:int;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{var list_<it.label.text>:Array=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
```

```
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( this.state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR
 start <ruleName>
public final function m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):void {
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   try {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       var _type:int = <ruleName>;
       var _channel:int = DEFAULT_TOKEN_CHANNEL;
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block>
       <ruleCleanUp()>
       this.state.type = _type;
       this.state.channel = _channel;
       <(ruleDescriptor.actions.after):execAction()>
<endif>
```

```
   }
   finally {
       <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
       <ruleScopeCleanUp()>
       <memoize()>
   }
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined
 lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
public override function mTokens():void {
   <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber>:int=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var alt<decisionNumber>:int=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
```

```
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description)
 ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
var cnt<decisionNumber>:int=0;
<decls>
<@preloop()>
loop<decisionNumber>:
do {
   var alt<decisionNumber>:int=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
     if ( cnt<decisionNumber> >= 1 ) break loop<decisionNumber>;
     <ruleBacktrackFailure()>
           throw new EarlyExitException(<decisionNumber>, input);
           <! Need to add support for earlyExitException
 debug hook !>
   }
   cnt<decisionNumber>++;
} while (true);
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock
```

```
/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
loop<decisionNumber>:
do {
   var alt<decisionNumber>:int=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
     break loop<decisionNumber>;
   }
} while (true);
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump
 to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> :
   <@prealt()>
   <it>
   break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label>=<labelType>(<endif>matchStream(input,<token>,FOLLOW_<token>_in_<ruleName><element
Index>)<if(label)>)<endif>; <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem)
 ::= <<
if (list_<label>==null) list_<label>=new Array();
list_<label>.push(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
```

```
<label> = input.LA(1);<\n>
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label>=<labelType>(input.LT(1));<\n>
<endif>
<endif>
if ( <s> ) {
  input.consume();
  <postmatchCode>
<if(!LEXER)>
  this.state.errorRecovery=false;
<endif>
  <if(backtracking)>this.state.failed=false;<endif>
}
else {
  <ruleBacktrackFailure()>
  <@mismatchedSetException()>
<if(LEXER)>
  throw recover(new MismatchedSetException(null,input));<\n>
<else>
  throw new MismatchedSetException(null,input);
   <! use following code to make it recover inline; remove throw mse;
  recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
  !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
var <label>Start:int = charIndex;
matchString(<string>); <checkRuleBacktrackFailure()>
```

```
<label> = CommonToken.createFromStream(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start, charIndex-1);
<else>
matchString(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label>=<labelType>(input.LT(1));<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label,
 elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*
* GMS: Note:  do not use post-decrement operator!  ASC produces bad code for exceptions in this case.
*      See: https://bugs.adobe.com/jira/browse/ASC-3625
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
state._fsp = state._fsp - 1;
<checkRuleBacktrackFailure()>
>>
```

```
/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
var <label>Start<elementIndex>:int = charIndex;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = CommonToken.createFromStream(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, charIndex-1);
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex)
 ::= <<
<if(label)>
var <label>Start<elementIndex>:int = charIndex;
match(EOF); <checkRuleBacktrackFailure()>
var <label>:<labelType> = CommonToken.createFromStream(input, EOF,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, charIndex-1);
<else>
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
```

```
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
   matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate
 (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
   <ruleBacktrackFailure()>
   throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   throw new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
   <! Need to add hook for noViableAltException() !>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's
```

```
 it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
   alt<decisionNumber>=<eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/**
 A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
```

```
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   throw new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
   <! Need to add hook for noViableAltException !>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
switch ( input.LA(<k>) ) {
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
   alt<decisionNumber>=<eotPredictsAlt>;
   break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n">
   {
   <targetState>
   }
   break;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input);
>>
```

```
cyclicDFACtor(dfa) ::= <<

dfa<dfa.decisionNumber> = new DFA(this, <dfa.decisionNumber>,
          "<dfa.description>",
          DFA<dfa.decisionNumber>_eot, DFA<dfa.decisionNumber>_eof,
 DFA<dfa.decisionNumber>_min,
          DFA<dfa.decisionNumber>_max, DFA<dfa.decisionNumber>_accept, DFA<dfa.decisionNumber>_special,
          DFA<dfa.decisionNumber>_transition<if(dfa.specialStateSTs)>,
DFA<dfa.decisionNumber>_specialStateTransition<endif>);

>>
/* Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<

private const DFA<dfa.decisionNumber>_eot:Array =
   DFA.unpackEncodedString("<dfa.javaCompressedEOT; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_eof:Array =
   DFA.unpackEncodedString("<dfa.javaCompressedEOF; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_min:Array
 =
   DFA.unpackEncodedString("<dfa.javaCompressedMin; wrap="\"+\n    \"">", true);
private const DFA<dfa.decisionNumber>_max:Array =
   DFA.unpackEncodedString("<dfa.javaCompressedMax; wrap="\"+\n    \"">", true);
private const DFA<dfa.decisionNumber>_accept:Array =
   DFA.unpackEncodedString("<dfa.javaCompressedAccept; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_special:Array =
   DFA.unpackEncodedString("<dfa.javaCompressedSpecial; wrap="\"+\n    \"">");
private const DFA<dfa.decisionNumber>_transition:Array = [
     <dfa.javaCompressedTransition:{s|DFA.unpackEncodedString("<s; wrap="\"+\n\"">")}; separator=",\n">
];
<if(dfa.specialStateSTs)>
   private function DFA<dfa.decisionNumber>_specialStateTransition(dfa:DFA, s:int, _input:IntStream):int {
     <if(LEXER)>
     var input:IntStream = _input;
     <endif>
     <if(PARSER)>
     var input:TokenStream = TokenStream(_input);
     <endif>
     <if(TREE_PARSER)>
     var input:TreeNodeStream
 = TreeNodeStream(_input);
     <endif>
   var _s:int = s;
     switch ( s ) {
     <dfa.specialStateSTs:{state |
     case <i0> : <! compressed special state numbers 0..n-1 !>
          <state>}; separator="\n">
     }
<if(backtracking)>
     if (this.state.backtracking>0) {this.state.failed=true; return -1;}<\n>
<endif>
     throw dfa.error(new NoViableAltException(dfa.description, <dfa.decisionNumber>, _s, input));
   }<\n>
<endif>

protected var dfa<dfa.decisionNumber>:DFA;  // initialized in constructor

>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
var LA<decisionNumber>_<stateNumber>:int = input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
var index<decisionNumber>_<stateNumber>:int = input.index;
input.rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)>
 <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
```

```
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o | ||<o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atomAsInt>"

/**
 Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atomAsInt>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber>\>=<lowerAsInt> &&
LA<decisionNumber>_<stateNumber>\<=<upperAsInt>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::=
"(input.LA(<k>)\>=<lowerAsInt> && input.LA(<k>)\<=<upperAsInt>)"

setTest(ranges) ::= "<ranges; separator=\"||\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected var <scope.name>_stack:Array = new Array();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected var <scope.name>_stack:Array = new Array();<\n>
<endif>
>>
```

```
returnStructName() ::= "<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope"

returnType()
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<returnStructName()>
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<asTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
var <label.label.text>:<ruleLabelType(referencedRule=label.referencedRule)> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>
```

```
/**
 Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public static class <returnType()> extends <if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope {
   <scope.attributes:{public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
};
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.name>:<it.type>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
<scope>_stack[<scope>_stack.length-<negIndex>-1].<attr.name>
<else>
<if(index)>
<scope>_stack[<index>].<attr.name>
<else>
<scope>_stack[<scope>_stack.length-1].<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
<scope>_stack[<scope>_stack.length-<negIndex>-1].<attr.name>
 =<expr>;
<else>
<if(index)>
<scope>_stack[<index>].<attr.name> =<expr>;
<else>
<scope>_stack[<scope>_stack.length-1].<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?<scope>.values.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.values.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.values.<attr.name>
 =<expr>;
<else>
<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.text:null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.charPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.tokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int(<scope>.text):0)"
```

```
ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=null?<labelType>(<scope>.start):null)"
ruleLabelPropertyRef_stop(scope,attr)
 ::= "(<scope>!=null?<labelType>(<scope>.stop):null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>!=null?<ASTLabelType>(<scope>.tree):null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(<scope>!=null?(input.tokenStream.toStringWithRange(
 input.treeAdaptor.getTokenStartIndex(<scope>.start),
 input.treeAdaptor.getTokenStopIndex(<scope>.start))):null)
<else>
(<scope>!=null?input.toStringWithRange(<scope>.start,<scope>.stop):null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "(<scope>!=null?<scope>.st:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(<scope>!=null?<scope>.type:0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(<scope>!=null?<scope>.lien:0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
   "(<scope>!=null?<scope>.charPositionInLine:0)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(<scope>!=null?<scope>.channel:0)"
lexerRuleLabelPropertyRef_index(scope,attr)
 ::=
   "(<scope>!=null?<scope>.tokenIndex:0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
   "(<scope>!=null?<scope>.text:null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
   "(<scope>!=null?int(<scope>.text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "<labelType>(retval.start)"
rulePropertyRef_stop(scope,attr) ::= "<labelType>(retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "<ASTLabelType>(retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.tokenStream.toStringWithRange(
 input.treeAdaptor.getTokenStartIndex(retval.start),
 input.treeAdaptor.getTokenStopIndex(retval.start))
<else>
input.toStringWithRange(retval.start,input.LT(-1))
<endif>
>>
```

```
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr)
 ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(charIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public
 static const <name>:BitSet = new BitSet([<words64:{<it>};separator=", ">]);<\n>
>>

codeFileExtension() ::= ".as"

true() ::= "true"
false() ::= "false"
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ActionScript/ActionScript.stg

No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to normal Java output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
group Dbg;

@outputFile.debugPreprocessor() ::= "#define ANTLR_DEBUG"

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.Runtime.Debug;
using IOException = System.IO.IOException;
>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
public static readonly string[]
```

```
 ruleNames = new string[] {
   "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n    ", separator=", ">
};<\n>
<endif>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
   private int ruleLevel = 0;
   public int RuleLevel {
 get { return ruleLevel; }
   }
   public void IncRuleLevel() { ruleLevel++; }
   public void DecRuleLevel() { ruleLevel--; }
<if(profile)>
   <ctorForProfilingRootGrammar()>
<else>
   <ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else> <! imported grammar !>
   public int RuleLevel {
 get { return <grammar.delegators:{g| <g:delegateName()>}>.RuleLevel; }
   }
   public void IncRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.IncRuleLevel(); }
   public void DecRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.DecRuleLevel(); }
   <ctorForDelegateGrammar()>
<endif>
<if(profile)>
override public bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
   ((Profiler)dbg).ExamineRuleMemoization(input,
 ruleIndex, <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
   return base.AlreadyParsedRule(input, ruleIndex);
}<\n>
override public void Memoize(IIntStream input,
                  int ruleIndex,
                  int ruleStartIndex)
{
   ((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
   base.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected bool EvalPredicate(bool result, string predicate)
{
   dbg.SemanticPredicate(result, predicate);
   return result;
}<\n>
>>
```

```
ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
public <name>(<inputStreamType> input)
   : this(input, DebugEventSocketProxy.DEFAULT_DEBUGGER_PORT, new RecognizerSharedState()) {
}

public <name>(<inputStreamType> input, int port, RecognizerSharedState state)

  : base(input, state) {
   <parserCtorBody()>
   <createListenerAndHandshake()>
   <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
   <@finally()>
}<\n>
>>

@parserCtorBody.initializeCyclicDFAs() ::= <<
InitializeCyclicDFAs(dbg);
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
public <name>(<inputStreamType> input) {
   this(input, new Profiler(null), new RecognizerSharedState());
}

public <name>(<inputStreamType> input, IDebugEventListener dbg, RecognizerSharedState state)
   : base(input, dbg, state) {
   Profiler p = (Profiler)dbg;
   p.setParser(this);
   <parserCtorBody()>
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state, this<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
   <@finally()>
}
<\n>
>>



/** Basically we don't want to
 set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
public <name>(<inputStreamType> input, IDebugEventListener dbg, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>)
   : base(input, dbg, state) {
```

```
   <parserCtorBody()>
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
}<\n>
>>

ctorForPredefinedListener() ::= <<
public <name>(<inputStreamType> input, IDebugEventListener dbg)
   : <@superClassRef>base(input, dbg, new RecognizerSharedState())<@end> {
<if(profile)>
   Profiler p = (Profiler)dbg;
   p.setParser(this);
<endif>
   <parserCtorBody()>
   <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
   <@finally()>
}<\n>
>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy
 proxy = new DebugEventSocketProxy(this, port, input.TreeAdaptor);
<else>
DebugEventSocketProxy proxy = new DebugEventSocketProxy(this, port, null);
<endif>
DebugListener = proxy;
try
{
   proxy.Handshake();
}
catch (IOException ioe)
{
   ReportError(ioe);
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

@rule.preamble() ::= <<
try {
 dbg.EnterRule(GrammarFileName, "<ruleName>");
 if ( RuleLevel==0 ) {dbg.Commence();}
 IncRuleLevel();
 dbg.Location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>);<\n>
>>
```

```
@lexer.debugInitializeCyclicDFAs() ::= "IDebugEventListener dbg"

@lexer.debugAddition() ::= ", dbg"

@genericParser.debugInitializeCyclicDFAs() ::= "IDebugEventListener dbg"

@genericParser.debugAddition() ::= ", dbg"

@rule.postamble() ::= <<
dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>);<\n>
}
finally {
  dbg.ExitRule(GrammarFileName, "<ruleName>");
  DecRuleLevel();
  if ( RuleLevel==0
 ) {dbg.Terminate();}
}<\n>
>>

@synpred.start() ::= "dbg.BeginBacktrack(state.backtracking);"

@synpred.stop() ::= "dbg.EndBacktrack(state.backtracking, success);"

// Common debug event triggers used by region overrides below

enterSubRule() ::=
  "try { dbg.EnterSubRule(<decisionNumber>);<\n>"

exitSubRule() ::=
  "} finally { dbg.ExitSubRule(<decisionNumber>); }<\n>"

enterDecision() ::=
  "try { dbg.EnterDecision(<decisionNumber>);<\n>"

exitDecision() ::=
  "} finally { dbg.ExitDecision(<decisionNumber>); }<\n>"

enterAlt(n) ::= "dbg.EnterAlt(<n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"
```

```
@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop()
 ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
   "dbg.RecognitionException(eee<decisionNumber>);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
   "dbg.Location(<it.line>,<it.pos>);"

@matchSet.mismatchedSetException() ::=
   "dbg.RecognitionException(mse);"

@dfaState.noViableAltException() ::= "dbg.RecognitionException(nvae_d<decisionNumber>s<stateNumber>);"

@dfaStateSwitch.noViableAltException() ::=
"dbg.RecognitionException(nvae_d<decisionNumber>s<stateNumber>);"

dfaDecision(decisionNumber,description) ::= <<
try

{
   isCyclicDecision = true;
   <super.dfaDecision(...)>
}
```

```
catch (NoViableAltException nvae)
{
  dbg.RecognitionException(nvae);
  throw nvae;
}
>>

@cyclicDFA.dbgCtor() ::= <<
  public DFA<dfa.decisionNumber>(BaseRecognizer recognizer, IDebugEventListener dbg) : this(recognizer)
  {
 this.dbg = dbg;
  }
>>

@cyclicDFA.debugMember() ::= <<
IDebugEventListener dbg;

>>

@cyclicDFA.errorMethod() ::= <<
public override void Error(NoViableAltException nvae)
{
  dbg.RecognitionException(nvae);
}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
EvalPredicate(<pred>,"<description>")
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/Dbg.stg

No license file was found, but licenses were detected in source scan.

documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/Perl5Target.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2006 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
/*
New style messages. This file contains the actual layout of the messages emitted by ANTLR.
The text itself is coming out of the languages/*stg files, according to the chosen locale.
This file contains the default format ANTLR uses.
*/

group antlr;

location(file, line, column) ::= "<file>(<line>,<column>)"

message(id, text) ::= "error <id> : <text>"

report(location, message, type) ::= "<location> : <type> <message.id> : <message.text>"

wantsSingleLineMessage() ::= "true"
```

Found
 in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/templates/messages/formats/vs2005.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/
group ASTDbg;

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor =
   new DebugTreeAdaptor(null,new CommonTreeAdaptor());
public void setTreeAdaptor(TreeAdaptor adaptor) {
   this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
}
public TreeAdaptor
 getTreeAdaptor() {
   return adaptor;
}<\n>
>>

parserCtorBody() ::= <<
>>

createListenerAndHandshake() ::= <<
<super.createListenerAndHandshake()>
>>

ctorForPredefinedListener() ::= <<
>>

@rewriteElement.pregen() ::= ""
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/C/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
<!protected TreeAdaptor adaptor = new CommonTreeAdaptor();<\n>!>
setTreeAdaptor: function(adaptor) {
   this.adaptor = adaptor;
   <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
},
getTreeAdaptor: function() {
   return this.adaptor;
},
>>

@returnScope.ruleReturnMembers()
 ::= <<
getTree: function() { return this.tree; }
>>
```

```
/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{var <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{var <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleTokenStream(this.adaptor,"token <it>");};
separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations()
 ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = this.adaptor.nil();<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
```

```
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.getTree());
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.getTree());
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

//
 Rewrite

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;
<rewriteCodeLabels()>
root_0 = this.adaptor.nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = this.adaptor.rulePostProcessing(root_0);
this.input.replaceChildren(this.adaptor.getParent(retval.start),
                      this.adaptor.getChildIndex(retval.start),
                     this.adaptor.getChildIndex(_last),
                     retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
```

```
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRule<rewriteElementType>Stream(this.adaptor,"token
<it>",<it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRule<rewriteElementType>Stream(this.adaptor,"token
<it>", list_<it>);};
   separator="\n"
>
<referencedRuleLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"token
<it>",<it>!=null?<it>.tree:null);};
    separator="\n"
>
<referencedRuleListLabels
   :{var stream_<it>=new org.antlr.runtime.tree.RewriteRuleSubtreeStream(this.adaptor,"token <it>",list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElements:{el
```

```
 | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext()}; separator="||">) ) {
   throw new org.antlr.runtime.tree.RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
var root_<treeLevel> = this.adaptor.nil();
<root:rewriteElement()>
<children:rewriteElement()>
this.adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements)
 ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
```

```
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
this.adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args)
 ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<createRewriteNodeFromElement(...)>, root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
this.adaptor.addChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
```

```
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>

rewriteRuleRefRoot(rule)
 ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
this.adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
this.adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = this.adaptor.becomeRoot(stream_<label>.nextNode(),
 root_<treeLevel>);<\n>
>>

createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel, args) !>
new <hetero>(<tokenType><if(args)>, <args; separator=", "><endif>)
```

```
<else>
this.adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.nextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
this.adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/JavaScript/AST.stg
No license file was found, but licenses were detected in source scan.

```
group Delphi implements ANTLRCore;

csharpTypeInitMap ::= [
 "int":"0",
 "uint":"0",
 "long":"0",
 "ulong":"0",
 "float":"0.0",
 "double":"0.0",
 "bool":"False",
 "byte":"0",
 "sbyte":"0",
 "short":"0",
 "ushort":"0",
 "char":"#0",
 "string":"''",
 "String":"''",
 default:"nil" // anything other than an atomic type
]

/** The overall file structure of a
 recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*  LEXER (Boolean): should we generate lexer code?
*  PARSER (Boolean): should we generate parser code?
*  TREE_PARSER (Boolean): should we generate tree parser code?
*  actionScope (String): 'lexer', 'parser', 'tree_parser' or custom scope
*  actions (HashMap):
*  docComment (String): document comment
*  recognizer (Object): recognizer class generator
*  name (String): name of grammar
*  tokens (HashMap<name: String, type: Integer>):
*  tokenNames:
*  rules:
*  cyclicDFAs:
*  bitsets:
*  buildTemplate (Boolean): should we generate a string template?
*  buildAST (Boolean): should we generate an AST?
*  rewriteMode (Boolean): are we rewriteing nodes?
*  profile (Boolean):
*  backtracking (Boolean): backtracking mode?
*  synpreds (): syntactic predicates
*  memoize (Boolean): should we memoize?
*  numRules (Integer): number of rules
*  fileName (String): fully qualified name of original .g
 file
*  ANTLRVersion (String): ANTLR version in Major.Minor.Build format
*  generatedTimestamp (String): date/time when the file is generated
*  trace (Boolean): should we trace input/output?
*  scopes:
*  superClass (String): name of base class, or empty string
*  literals:
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
unit <name>;

{$HINTS OFF}

// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<actions.(actionScope).header>

interface

<@imports>
uses<\n>
<@end>
 <actions.(actionScope).usesInterface>
<if(TREE_PARSER)>
 Antlr.Runtime.Tree,<\n>
<endif>
 Antlr.Runtime,
 Antlr.Runtime.Collections,
 Antlr.Runtime.Tools;

<docComment>
<recognizer>
>>

/** Generates
 source code for the lexer class
* grammar (Grammar object)
*/
lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="Lexer") ::= <<
```

```
type
 I<grammar.recognizerName> = interface(I<@superClassName><superClass><@end>)
 end;

 T<grammar.recognizerName> = class(T<@superClassName><superClass><@end>,
I<grammar.recognizerName>)
 strict private
   FCnt: array [0..<grammar.numberOfDecisions>] of Byte;
   FLA: array [0..<grammar.numberOfDecisions>, 0..255] of Integer;
   FException: ERecognitionException;
   procedure InitializeCyclicDFAs;
 <cyclicDFAs:cyclicDFADeclaration()>
 public
   const
     <tokens:{<it.name> = <it.type>;}; separator="\n">
   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 strict private
   <actions.(actionScope).memberDeclarations>
 public
   // delegates
   <grammar.delegates: {g|<g:delegateName()>: I<superClass>; {<g.recognizerName>}}; separator="\n">
 public
    // delegators
   <grammar.delegators: {g|<g:delegateName()>: Pointer; {<g.recognizerName>}}; separator="\n">
   <last(grammar.delegators):{g|gParent: Pointer; {<g.recognizerName>}}>
 protected
   { IBaseRecognizer }
   function GetGrammarFileName: String; override;
<if(filterMode)>
   function AlreadyParsedRule(const Input: IIntStream;
     const RuleIndex: Integer): Boolean; override;
   procedure Memoize(const Input: IIntStream; const RuleIndex,
     RuleStartIndex: Integer); override;
 protected
   { ILexer }
   function NextToken: IToken; override;<\n>
<endif>
 protected
   { ILexer }
   procedure DoTokens; override;
 public
   constructor Create; overload;
   constructor Create(const AInput: ICharStream<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>); overload;
   constructor Create(const AInput: ICharStream; const AState: IRecognizerSharedState<grammar.delegators:{g|;
const A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>);
 overload;
```

```
   <rules: {r | <if(!r.ruleDescriptor.isSynPred)><lexerRuleDeclaration(r)><endif>}>
   <synpreds:{p | <lexerSynpredDeclaration(p)>}; separator="\n">
 end;

implementation

uses
 <grammar.delegates: {g|<g.recognizerName>,}; separator="\n">
 <grammar.delegators: {g|<g.recognizerName>,}; separator="\n">
 <actions.(actionScope).usesImplementation>
 SysUtils,
 StrUtils,
 Math;

{ T<grammar.recognizerName> }

constructor T<grammar.recognizerName>.Create;
begin
 InitializeCyclicDFAs;
end;

constructor T<grammar.recognizerName>.Create(const AInput: ICharStream<grammar.delegators:{g|; const
A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>);
begin
 Create(AInput, nil<grammar.delegators:{g|, A<g:delegateName()>}>);
end;

constructor T<grammar.recognizerName>.Create(const AInput: ICharStream; const AState:
IRecognizerSharedState<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>);
begin
 inherited Create(AInput,
 AState);
 InitializeCyclicDFAs; { TODO: Necessary in Delphi??? Not removed yet. }
 <if(memoize)>
 <if(grammar.grammarIsRoot)>
 State.RuleMemoCount := <numRules>+1;<\n> <! index from 1..n !>
 <endif>
 <endif>
 <grammar.directDelegates:
  {g|<g:delegateName()> := T<g.recognizerName>.Create(AInput, State<trunc(g.delegators):{p|,
<p:delegateName()>}>, Self);}; separator="\n">
 <grammar.delegators:
  {g|<g:delegateName()> := Pointer(A<g:delegateName()>);}; separator="\n">
 <last(grammar.delegators):{g|gParent := Pointer(A<g:delegateName()>);}>
 <actions.(actionScope).memberInitializations>
end;
<actions.(actionScope).memberImplementations>
```

```
function T<grammar.recognizerName>.GetGrammarFileName: String;
begin
 Result := '<fileName>';
end;

<if(filterMode)>
<filteringNextToken()>
<endif>

<rules; separator="\n\n">
<synpreds:{p | <lexerSynpred(p)>}>

procedure T<grammar.recognizerName>.InitializeCyclicDFAs;
begin
 <cyclicDFAs:{dfa | FDFA<dfa.decisionNumber> :=
TDFA<dfa.decisionNumber>.Create(Self<@debugAddition()>);};
 separator="\n">
 <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>FDFA<dfa.decisionNumber>.SpecialStateTransitionHandler :=
DFA<dfa.decisionNumber>_SpecialStateTransition;<endif>}; separator="\n">
end;

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
end.>>

lexerRuleDeclaration(rule) ::= <<
procedure m<rule.ruleName>(<rule.ruleDescriptor.parameterScope:parameterScope(scope=rule)>);<\n>
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
function T<grammar.recognizerName>.NextToken: IToken;
var
 M: Integer;
begin
 while (True) do
 begin
   if (Input.LA(1) = Integer(cscEOF)) then
     Exit(TToken.EOF_TOKEN);

   State.Token :=
 nil;
   State.Channel := TToken.DEFAULT_CHANNEL;
   State.TokenStartCharIndex := Input.Index;
   State.TokenStartCharPositionInLine := Input.CharPositionInLine;
   State.TokenStartLine := Input.Line;
   State.Text := '';
   try
     M := Input.Mark();
     State.Backtracking := 1; <! means we won't throw slow exception !>
     State.Failed := False;
     mTokens();
     State.Backtracking := 0;
<!
     mTokens backtracks with synpred at backtracking==2
           and we set the synpredgate to allow actions at level 1.
!>
     if (State.Failed) then
     begin
       Input.Rewind(M);
       Input.Consume; <! // advance one char and try again !>
     end
     else
     begin
       Emit;
       Exit(State.Token);
     end;
   except
     on RE: ERecognitionException do
     begin
       // shouldn't happen in backtracking mode, but...
       ReportError(RE);
       Recover(RE);
     end;
   end;
 end;
end;

function T<grammar.recognizerName>.AlreadyParsedRule(const
 Input: IIntStream;
 const RuleIndex: Integer): Boolean;
begin
 if (State.Backtracking > 1) then
   Result := inherited AlreadyParsedRule(Input, RuleIndex)
 else
   Result := False;
end;

procedure T<grammar.recognizerName>.Memoize(const Input: IIntStream; const RuleIndex,
 RuleStartIndex: Integer);
begin
```

```
 if (State.Backtracking > 1) then
   inherited Memoize(Input, RuleIndex, RuleStartIndex);
end;

>>

filteringActionGate() ::= "(State.Backtracking = 1)"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
             bitsets, inputStreamType, superClass, filterMode,
             ASTLabelType="ANTLRInterface", labelType, members, rewriteElementType) ::= <<
type
 <rules: {r | <genericParserRuleReturnType(rule=r, ruleDescriptor=r.ruleDescriptor)>}>
 I<grammar.recognizerName> = interface(I<@superClassName><superClass><@end>)
   <rules: {r | <genericParserRuleInterface(rule=r, ruleDescriptor=r.ruleDescriptor)>}>
  end;

 T<grammar.recognizerName> = class(T<@superClassName><superClass><@end>,
I<grammar.recognizerName>)
<if(grammar.grammarIsRoot)>
 public
   const
     TOKEN_NAMES: array [0..<length(tokenNames)>+3] of String = (
       '\<invalid>',
       '\<EOR>',
       '\<DOWN>',
       '\<UP>',
       <tokenNames; separator=",\n">);<\n>
<endif>
 public
   const
     <tokens:{<it.name> = <it.type>;}; separator="\n">
 public
   // delegates
   <grammar.delegates: {g|<g:delegateName()>: I<superClass>; {<g.recognizerName>}}; separator="\n">
 public
   // delegators
   <grammar.delegators: {g|<g:delegateName()>: Pointer; {<g.recognizerName>}}; separator="\n">
   <last(grammar.delegators):{g|gParent: Pointer; {<g.recognizerName>}}>

   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeDeclaration(scope=it)><endif>}>
<@members>
   <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
 public
   constructor Create(const AInput: <inputStreamType><grammar.delegators:{g|;
 const A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>); overload;
   constructor Create(const AInput: <inputStreamType>; const AState:
```

```
IRecognizerSharedState<grammar.delegators:{g|; const A<g:delegateName()>:
IBaseRecognizer{<g.recognizerName>}}>); overload;
<@end>
 protected
   { IBaseRecognizer }
   function GetTokenNames: TStringArray; override;
   function GetGrammarFileName: String; override;
 strict private
   <actions.(actionScope).memberDeclarations>
 <rules: {r | <genericParserRuleDeclaration(rule=r, ruleDescriptor=r.ruleDescriptor)>}>

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
   // Delegated rules
   <grammar.delegatedRules:{ruleDescriptor| <delegatedRuleDeclaration(ruleDescriptor)>}>

   <synpreds:{p | <synpredDeclaration(p)>}; separator="\n">
 <cyclicDFAs:cyclicDFADeclaration()>
 strict private
   FException: ERecognitionException;
   FLA: array [0..<grammar.numberOfDecisions>,
 0..255] of Integer;
   FCnt: array [0..<grammar.numberOfDecisions>] of Byte;
   procedure InitializeCyclicDFAs;
<if(bitsets)>
 public
   class var
     <bitsets:bitsetDecl(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>})>
 public
   class procedure InitializeBitsets; static;<\n>
<endif>
 end;

implementation

uses
 <grammar.delegates: {g|<g.recognizerName>,}; separator="\n">
 <grammar.delegators: {g|<g.recognizerName>,}; separator="\n">
 <actions.(actionScope).usesImplementation>
 SysUtils,
 StrUtils,
 Math;

{ T<grammar.recognizerName> }

constructor T<grammar.recognizerName>.Create(const AInput: <inputStreamType><grammar.delegators:{g|; const
A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>);
begin
```

```
 Create(AInput, TRecognizerSharedState.Create<grammar.delegators:{g|, A<g:delegateName()>}>);
end;

constructor T<grammar.recognizerName>.Create(const AInput: <inputStreamType>;
 const AState: IRecognizerSharedState<grammar.delegators:{g|;
 const A<g:delegateName()>: IBaseRecognizer{<g.recognizerName>}}>);
begin
 inherited Create(AInput, AState);
 <@membersConstructor>
 <@end>
 <parserCtorBody()>
 <grammar.directDelegates:{g|<g:delegateName()> := T<g.recognizerName>.Create(Input,
State<trunc(g.delegators):{p|, <p:delegateName()>}>, Self);}; separator="\n">
 <grammar.indirectDelegates:{g | <g:delegateName()> := <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
 <last(grammar.delegators):{g|gParent := Pointer(A<g:delegateName()>);}>
 <rules: {r | <ruleAttributeScopeInit(scope=r.ruleDescriptor.ruleScope)>}>
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <actions.(actionScope).memberInitializations>
end;
<actions.(actionScope).memberImplementations>

<grammar.delegatedRules:{ruleDescriptor| <delegatedRuleImplementation(ruleDescriptor)>}; separator="\n">
procedure T<grammar.recognizerName>.InitializeCyclicDFAs;
begin
 <cyclicDFAs:{dfa | FDFA<dfa.decisionNumber>
 := TDFA<dfa.decisionNumber>.Create(Self);}; separator="\n">
 <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>FDFA<dfa.decisionNumber>.SpecialStateTransitionHandler :=
DFA<dfa.decisionNumber>_SpecialStateTransition;<endif>}; separator="\n">
end;

<if(bitsets)>
class procedure T<grammar.recognizerName>.InitializeBitsets;
begin
 <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>
end;
<endif>

<@membersImplementation>
<@end>

function T<grammar.recognizerName>.GetTokenNames: TStringArray;
var
 I: Integer;
begin
 SetLength(Result,Length(T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES));
 for I := 0 to Length(T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES) - 1 do
   Result[I] := T<grammar.composite.rootGrammar.recognizerName>.TOKEN_NAMES[I];
```

```
end;

function T<grammar.recognizerName>.GetGrammarFileName: String;
begin
 Result := '<fileName>';
end;

<rules; separator="\n\n">
<synpreds:{p | <synpred(p)>}>

<cyclicDFAs:cyclicDFA()>
 <! dump tables for all DFA !>
<if(bitsets)>
initialization
 T<grammar.recognizerName>.InitializeBitsets;<\n>
<endif>
end.>>

delegatedRuleDeclaration(ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<returnType()>;<\n>
<else>
procedure <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);<\n>
<endif>
<endif>
>>

delegatedRuleImplementation(ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
function
T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
): I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function
T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
):
 <returnType()>;<\n>
<else>
procedure
T<grammar.recognizerName>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>
);<\n>
<endif>
```

```
<endif>
begin
 <if(ruleDescriptor.hasReturnValue)>Result :=<endif>
T<ruleDescriptor.grammar.recognizerName>(<ruleDescriptor.grammar:delegateName()>.Implementor).<ruleDescri
ptor.name>(<ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", ">);
end;

>>

parserCtorBody() ::= <<
InitializeCyclicDFAs;
<if(memoize)>
<if(grammar.grammarIsRoot)>
State.RuleMemoCount := <length(grammar.allImportedRules)>+1;<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators: {g|<g:delegateName()> := Pointer(A<g:delegateName()>);}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType, superClass="Parser",
labelType="Token", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="Token", ...)>
>>

/** How
 to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start "<ruleName>"
procedure
T<grammar.recognizerName>.<ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
;
var
```

```
  Alt: array [0..<grammar.numberOfDecisions>] of Integer;
 <ruleLabelDefVars()>
begin
 <ruleLabelDefs()>
<if(trace)>
 TraceIn('<ruleName>_fragment', <ruleDescriptor.index>);
 try
   <block>
 finally
   TraceOut('<ruleName>_fragment', <ruleDescriptor.index>);
 end;
<else>
 <block>
<endif>
end;
// $ANTLR end "<ruleName>"
>>

synpredDecls(name) ::= <<
SynPredPointer <name>;<\n>
>>

synpred(name) ::= <<

function T<grammar.recognizerName>.<name>: Boolean;
var
 Start: Integer;
 Success: Boolean;
begin
 State.Backtracking := State.Backtracking + 1;
 <@start()>
 Start := Input.Mark;
 try
   <name>_fragment(); // can never throw exception
 except
   on RE: ERecognitionException do
     WriteLn('Impossible: ' + RE.ToString);
 end;
 Success := not State.Failed;
 Input.Rewind(Start);
 <@stop()>
 State.Backtracking := State.Backtracking - 1;
 State.Failed := False;
 Result := Success;
end;<\n>
>>

lexerSynpred(name) ::= <<
```

```
<synpred(name)>
>>

lexerSynpredDeclaration(name)
 ::= <<
function <name>: Boolean;
procedure <name>_fragment;
>>

synpredDeclaration(name) ::= <<
function <name>: Boolean;
procedure <name>_fragment;
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ((State.Backtracking > 0) and AlreadyParsedRule(Input, <ruleDescriptor.index>)) then
 Exit(<ruleReturnValue()>);
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)><\n>if (State.Failed) then Exit(<ruleReturnValue()>);<\n><endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (State.Backtracking > 0) then
begin
 State.Failed := True;
 Exit(<ruleReturnValue()>);
end;<endif>
>>

genericParserRuleDeclaration(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.isSynPred)>
<else>
<ruleAttributeScopeDeclaration(scope=ruleDescriptor.ruleScope)>
<returnScopeDeclaration(scope=ruleDescriptor.returnScope)>
public
<if(ruleDescriptor.hasMultipleReturnValues)>
  function <rule.ruleName>: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
 function <rule.ruleName>: <returnType()>;<\n>
<else>
 procedure <rule.ruleName>;<\n>
```

```
<endif>
<endif>
<endif>
>>

genericParserRuleInterface(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.isSynPred)>
<else>
<if(ruleDescriptor.hasMultipleReturnValues)>
function <rule.ruleName>: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function <rule.ruleName>: <returnType()>;<\n>
<else>
procedure <rule.ruleName>;<\n>
<endif>
<endif>
<endif>
>>

genericParserRuleReturnType(rule, ruleDescriptor) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(ruleDescriptor.isSynPred)>
<else>
I<returnType()> = interface(I<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope)
end;<\n>
<endif>
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize)
 ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
(* <fileName>:<description> *)
<if(ruleDescriptor.hasMultipleReturnValues)>
function T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
I<returnType()>;
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
function T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<returnType()>;
<else>
```

```
procedure
T<grammar.recognizerName>.<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);
<endif>
<endif>

var
<ruleDescriptor.actions.vars>
 Locals: TLocalStorage;
<if(ruleDescriptor.hasMultipleReturnValues)>
 RetVal: I<returnType()>;<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
 RetVal: <returnType()>;<\n>
<else>
<endif>
<endif>
 Alt: array [0..<grammar.numberOfDecisions>] of Integer;
  <ruleDeclarationVars()>
 <ruleLabelDefVars()>
begin
 Locals.Initialize;
 try
   <if(trace)>TraceIn('<ruleName>', <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   try
     try
       <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
<if(exceptions)>
       <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
       <actions.(actionScope).rulecatch>
<else>
     except
       on RE: ERecognitionException do
       begin
         ReportError(RE);
         Recover(Input,RE);
         <@setErrorReturnValue()>
       end;<\n>
```

```
<endif>
<endif>
<endif>
     end;
   finally
     <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
     <memoize()>
     <ruleScopeCleanUp()>
     <finally>
   end;
   <@postamble()>
  finally
   Locals.Finalize;
 end;
 Exit(<ruleReturnValue()>);
end;
// $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
   <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal := T<returnType()>.Create;
RetVal.Start := Input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name> := <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex := Input.Index();
<endif>
>>

ruleDeclarationVars() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name>: <a.type>;
}>
<endif>
<if(memoize)>
```

```
<ruleDescriptor.name>_StartIndex: Integer;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>Stack.Push(T<it>Scope.Create);}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>Stack.Push(T<it.name>Scope.Create);}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>Stack.Pop();};
 separator="\n">
<ruleDescriptor.ruleScope:{<it.name>Stack.Pop;}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]:{<it.label.text> := nil;}; separator="\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{list_<it.label.text> := nil;}; separator="\n">
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll.label.text> := nil;}; separator="\n">
>>

ruleLabelDefVars() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]:{<it.label.text>: I<labelType>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{list_<it.label.text>: IList\<IANTLRInterface\>;};
separator="\n">
<ruleDescriptor.ruleLabels:ruleLabelDefVar(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll.label.text>: <ruleLabelType(referencedRule=ll.referencedRule)>;};
separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
  ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<it.label.text> := nil;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{List_<it.label.text> := nil;}; separator="\n"
>
>>

lexerRuleLabelDefDeclarations() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<it.label.text>: I<labelType>;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{List_<it.label.text>: IList;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
RetVal
<endif>
<else>
<! nil !>
<endif>
<endif>
>>

ruleCleanUp()
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
RetVal.Stop := Input.LT(-1);
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if (State.Backtracking > 0) then
 Memoize(Input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex);
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
```

```
<ruleDescriptor.parameterScope>
procedure
T<grammar.recognizerName>.m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>);
var
 <ruleDescriptor.actions.vars>
 Locals: TLocalStorage;
 TokenType, Channel: Integer;
 Alt: array [0..<grammar.numberOfDecisions>] of Integer;
 <lexerRuleLabelDefDeclarations()>
begin
 Locals.Initialize;
 try
   <ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
   <if(trace)>TraceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
    <ruleDeclarations()>
   try
<if(nakedBlock)>
     <ruleMemoization(name=ruleName)>
     <lexerRuleLabelDefs()>
     <ruleDescriptor.actions.init>
     <block><\n>
<else>
     TokenType := <ruleName>;
     Channel := DEFAULT_TOKEN_CHANNEL;
     <ruleMemoization(name=ruleName)>
     <lexerRuleLabelDefs()>
     <ruleDescriptor.actions.init>
     <block>
     <ruleCleanUp()>
     State.TokenType := TokenType;
     State.Channel := Channel;
     <(ruleDescriptor.actions.after):execAction()>
<endif>
   finally
     <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
     <ruleScopeCleanUp()>
     <memoize()>
   end;
 finally
   Locals.Finalize;
 end;
end;
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
```

```
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
procedure T<grammar.recognizerName>.mTokens;
var
 Alt: array [0..<grammar.numberOfDecisions>]
 of Integer;
begin
 <block>
end;

procedure T<grammar.recognizerName>.DoTokens;
begin
 mTokens;
end;
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
Alt[<decisionNumber>] := <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
case Alt[<decisionNumber>] of
 <alts:altSwitchCase()>
end;
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
(* <fileName>:<description> *)
Alt[<decisionNumber>] := <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
case Alt[<decisionNumber>] of
 <alts:altSwitchCase()>
end;
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
(* <fileName>:<description> *)
```

```
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/**
 A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
(* <fileName>:<description> *)
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
(* <fileName>:<description> *)
FCnt[<decisionNumber>] := 0;
<decls>
<@preloop()>
while (True) do
begin
 Alt[<decisionNumber>] := <maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 case Alt[<decisionNumber>] of
   <alts:altSwitchCase()>
 else
   begin
     if (FCnt[<decisionNumber>] >= 1) then
       Break;
     <ruleBacktrackFailure()>
     raise EEarlyExitException.Create(<decisionNumber>, Input);
     <@earlyExitException()>
   end;
 end;
 Inc(FCnt[<decisionNumber>]);
end;
<@postloop()>
>>

positiveClosureBlockSingleAlt
 ::= positiveClosureBlock
```

```
/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
(* <fileName>:<description> *)
<decls>
<@preloop()>
while (True) do
begin
 Alt[<decisionNumber>] := <maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 case Alt[<decisionNumber>] of
   <alts:altSwitchCase()>
 else
   Break;
 end;
end;
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
<i>:
 <@prealt()>
 <it><\n>
>>

/** An alternative is just a list of elements;
 at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
(* <fileName>:<description> *)
begin
 <@declarations()>
 <elements:element()>
 <rew>
 <@cleanup()>
end;
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label> := <endif>Match(Input, <token>,
FOLLOW_<token>_in_<ruleName><elementIndex>)<if(label)> as
I<labelType><endif>;<\n><checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> = nil) then list_<label> := TList\<IANTLRInterface\>.Create;
list_<label>.Add(<elem>);<\n>
>>

/** match a character
 */
charRef(char,label) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
```

```
<endif>
MatchRange(<a>, <b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label> := Input.LA(1);<\n>
<else>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
<endif>
if (<s>) then
begin
 Input.Consume;
 <postmatchCode>
 <if(!LEXER)>
 State.ErrorRecovery := False;<endif>
 <if(backtracking)>State.Failed := False;<endif>
end
else
begin
 <ruleBacktrackFailure()>
 FException := EMismatchedSetException.Create(nil, Input);
 <@mismatchedSetException()>
<if(LEXER)>
 Recover(FException);
 raise FException;<\n>
<else>
 raise FException;
 <! use following code to make it recover inline; remove throw mse;
 RecoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
  !>
<endif>
end;<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
```

```
Locals.AsInteger['<label>Start'] := CharIndex;
Match(<string>); <checkRuleBacktrackFailure()>
<label> := TCommonToken.Create(Input, TToken.INVALID_TOKEN_TYPE, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start'], CharIndex-1);
<else>
Match(<string>); <checkRuleBacktrackFailure()>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> := Input.LA(1);<\n>
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label,
 elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
<label> := <if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
<else>
<if(scope)>T<scope.recognizerName>(IANTLRObject(<scope:delegateName()>).Implementor).<endif><rule.name
>(<args; separator=", ">);<\n>
```

```
<endif>
State.FollowingStackPointer := State.FollowingStackPointer - 1;
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer
 rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
Locals.AsInteger['<label>Start<elementIndex>'] := CharIndex;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> := TCommonToken.Create(Input, TToken.INVALID_TOKEN_TYPE, TToken.DEFAULT_CHANNEL,
 Locals.AsInteger['<label>Start<elementIndex>'], CharIndex - 1);
<else>
<if(scope)>(<scope:delegateName()>.Implementor as T<scope.recognizerName>).<endif>m<rule.name>(<args;
separator=", ">); <checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
Locals.AsInteger['<label>Start<elementIndex>']
 := CharIndex;
Match(EOF); <checkRuleBacktrackFailure()>
Locals['<label>'] := TCommonToken.Create(Input, EOF, TToken.DEFAULT_CHANNEL,
Locals.AsInteger['<label>Start<elementIndex>'], CharIndex-1);
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
```

```
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if (Input.LA(1) = TToken.DOWN) then
begin
 Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
end;
<else>
Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
<children:element()>
Match(Input, TToken.UP, nil);<\n><checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description)
 ::= <<
if (not (<evalPredicate(...)>)) then
begin
 <ruleBacktrackFailure()>
 raise EFailedPredicateException.Create(Input, '<ruleName>', '<description>');
end;<\n>
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
begin
<if(eotPredictsAlt)>
 Alt[<decisionNumber>] := <eotPredictsAlt>;<\n>
<else>
 <ruleBacktrackFailure()>
 raise ENoViableAltException.Create('<description>', <decisionNumber>, <stateNumber>, Input);<\n>
<endif>
end;
>>
```

```
/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>]
 := Input.LA(<k>);<\n>
<edges; separator="\nelse ">;
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(<k>);
<edges; separator="\nelse ">;<\n>
<if(eotPredictsAlt)>
<if(!edges)>
Alt[<decisionNumber>] := <eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else
begin
 Alt[<decisionNumber>] := <eotPredictsAlt>;
end;<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "Alt[<decisionNumber>] := <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter
 to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> and (<predicates>)<endif>) then
begin
 <targetState>
end <! no ; here !>
>>
```

```
// Fixed DFA (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
 <edges; separator="\n">
else
 begin
<if(eotPredictsAlt)>
   Alt[<decisionNumber>] := <eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   <@noViableAltException()>
   raise ENoViableAltException.Create('<description>', <decisionNumber>, <stateNumber>, Input);<\n>
<endif>
 end;
end;<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
 <edges; separator="\n">
end;<\n>
>>

dfaLoopbackStateSwitch(k,
 edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
case Input.LA(<k>) of
 <edges; separator="\n"><\n>
<if(eotPredictsAlt)>
else
 Alt[<decisionNumber>] := <eotPredictsAlt>;<\n>
<endif>
end;<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{<it>}; separator=",\n">:
 begin
   <targetState>
 end;
>>

// Cyclic DFA

/** The code to initiate execution of a cyclic DFA; this is used
```

```
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
Alt[<decisionNumber>] := FDFA<decisionNumber>.Predict(Input);
>>

/* Dump DFA tables.
*/
cyclicDFADeclaration(dfa) ::= <<
strict protected
 type
   TDFA<dfa.decisionNumber> = class(TDFA)
   protected
     { IDFA }
     function Description: String; override;
   public
     constructor Create(const ARecognizer: IBaseRecognizer);
   end;
 var
   FDFA<dfa.decisionNumber>:
 IDFA;
<if(dfa.specialStateSTs)>
strict protected
 function DFA<dfa.decisionNumber>_SpecialStateTransition(const DFA: IDFA; S: Integer;
   const AInput: IIntStream): Integer;<endif>
>>

cyclicDFA(dfa) ::= <<
{ T<grammar.recognizerName>.TDFA<dfa.decisionNumber> }

constructor T<grammar.recognizerName>.TDFA<dfa.decisionNumber>.Create(const ARecognizer:
IBaseRecognizer);
const
 DFA<dfa.decisionNumber>_EOT = '<dfa.javaCompressedEOT; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_EOF = '<dfa.javaCompressedEOF; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_MIN = '<dfa.javaCompressedMin; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_MAX = '<dfa.javaCompressedMax; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_ACCEPT = '<dfa.javaCompressedAccept; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_SPECIAL = '<dfa.javaCompressedSpecial; wrap="'+\n    '">';
 DFA<dfa.decisionNumber>_TRANSITION: array [0..<length(dfa.javaCompressedTransition)>-1] of String = (
   <dfa.javaCompressedTransition:{s|'<s;
 wrap="'+\n'">'}; separator=",\n">);
begin
 inherited Create;
 Recognizer := ARecognizer;
 DecisionNumber := <dfa.decisionNumber>;
 EOT := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_EOT);
 EOF := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_EOF);
 Min := TDFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_MIN);
 Max := TDFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_MAX);
 Accept := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_ACCEPT);
 Special := TDFA.UnpackEncodedString(DFA<dfa.decisionNumber>_SPECIAL);
 Transition := TDFA.UnpackEncodedStringArray(DFA<dfa.decisionNumber>_TRANSITION);
end;

function T<grammar.recognizerName>.TDFA<dfa.decisionNumber>.Description: String;
begin
 Result := '<dfa.description>';
end;<\n>
<if(dfa.specialStateSTs)>
function T<grammar.recognizerName>.DFA<dfa.decisionNumber>_SpecialStateTransition(const DFA: IDFA; S:
Integer;
 const AInput: IIntStream): Integer;
var
 Locals: TLocalStorage;
  <if(LEXER)>
 Input: IIntStream;
 <endif>
 <if(PARSER)>
 Input: ITokenStream;
 <endif>
 <if(TREE_PARSER)>
 Input: ITreeNodeStream;
 <endif>
 _S: Integer;
 NVAE: ENoViableAltException;
begin
 Result := -1;
 Locals.Initialize;
 try
   <if(LEXER)>
   Input := AInput;
   <endif>
   <if(PARSER)>
   Input := AInput as ITokenStream;
   <endif>
   <if(TREE_PARSER)>
   Input := AInput as ITreeNodeStream;
   <endif>
   _S := S;
   case S of
     <dfa.specialStateSTs:{state | <i0>: begin<! compressed special state numbers 0..n-1 !>
    <state> <\n>   end;}; separator="\n">
   end;
<if(backtracking)>
```

```
   if (State.Backtracking > 0) then
   begin
     State.Failed := True;
     Exit(-1);
   end;<\n>
<endif>
   NVAE := ENoViableAltException.Create(DFA.Description, <dfa.decisionNumber>, _S, Input);
   DFA.Error(NVAE);
   raise NVAE;
 finally
   Locals.Finalize;
 end;
end;<\n>
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big
 switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
FLA[<decisionNumber>,<stateNumber>] := Input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
Locals.AsInteger['index<decisionNumber>_<stateNumber>'] := Input.Index;
Input.Rewind;<\n>
<endif>
S := -1;
<edges; separator="\nelse ">;
<if(semPredState)> <! return input cursor to state before we rewound !>
Input.Seek(Locals.AsInteger['index<decisionNumber>_<stateNumber>']);<\n>
<endif>
if (S >= 0) then
 Exit(S);
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ((<labelExpr>)<if(predicates)> and (<predicates>)<endif>) then
 S := <targetStateNumber>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates)
```

```
 ::= <<
S := <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "((<first(operands)>)<rest(operands):{o | or (<o>)}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "FLA[<decisionNumber>,<stateNumber>] = <atomAsInt>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "Input.LA(<k>) = <atomAsInt>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
((FLA[<decisionNumber>,<stateNumber>] \>= <lowerAsInt>) and (FLA[<decisionNumber>,<stateNumber>] \<=
<upperAsInt>))
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(Input.LA(<k>) \>=
<lowerAsInt>) and (Input.LA(<k>)
 \<= <upperAsInt>)"

setTest(ranges) ::= "<ranges; separator=\") or (\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<scope.name>Stack := TStackList\<I<scope.name>Scope\>.Create;<\n>
<endif>
>>

globalAttributeScopeDeclaration(scope) ::= <<
<if(scope.attributes)>
strict protected
 type
   I<scope.name>Scope = interface(IANTLRObject)
   end;
```

```
  T<scope.name>Scope = class(TANTLRObject, I<scope.name>Scope)
  protected
    <scope.attributes:{<it.name>: <it.type>;}; separator="\n">
  end;
strict protected
 <scope.name>Stack: IStackList\<I<scope.name>Scope\>;
<endif>
>>

ruleAttributeScopeDeclaration(scope) ::= <<
<if(scope.attributes)>
strict protected
 type
  I<scope.name>Scope = interface(IANTLRObject)
  end;
  T<scope.name>Scope = class(TANTLRObject, I<scope.name>Scope)
  protected
    <scope.attributes:{<it.name>: <it.type>;}; separator="\n">
  end;
strict protected
 <scope.name>Stack: IStackList\<I<scope.name>Scope\>;
<endif>
>>

ruleAttributeScope(scope)
 ::= <<
<! protected Stack <scope.name>Stack = new Stack();<\n> !>
>>

ruleAttributeScopeInit(scope) ::= <<
<if(scope)>
<scope.name>Stack := TStackList\<I<scope.name>Scope\>.Create;<\n>
<endif>
>>

returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor:returnStructName()>
<! I<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope !>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
<! Pointer/void !>
<endif>
<endif>
```

```
>>

/** Generate the C# type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
I<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type;
 if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<csharpTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> := <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

ruleLabelDefVar(label) ::= <<
<label.label.text>: <ruleLabelType(referencedRule=label.referencedRule)>;
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
{ T<ruleDescriptor:returnStructName()> }

<scope.attributes:{public <it.decl>;}; separator="\n">
<@ruleReturnMembers()>
```

```
<endif>
>>

returnScopeDeclaration(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public
 type
   T<ruleDescriptor:returnStructName()> =
class(T<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope,
 I<ruleDescriptor:returnStructName()>)
   <scope.attributes:{public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
   end;
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> := <expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
(<scope>Stack[<scope>Stack.Count-<negIndex>-1] as T<scope>Scope).<attr.name>
<else>
<if(index)>
(<scope>Stack[<index>] as T<scope>Scope).<attr.name>
((<scope>_scope)<scope>_stack[<index>]).<attr.name>
<else>
(<scope>Stack.Peek.Implementor as T<scope>Scope).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
(<scope>Stack[<scope>Stack.Count-<negIndex>-1] as T<scope>Scope).<attr.name> := <expr>;<\n>
<else>
<if(index)>
(<scope>Stack[<index>] as T<scope>Scope).<attr.name> := <expr>;<\n>
<else>
(<scope>Stack.Peek.Implementor as T<scope>Scope).<attr.name>
 := <expr>;<\n>
<endif>
<endif>
>>
```

```
/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>Stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(IfThen(Assigned(<scope>),Def(<scope>).<attr.name>,<initValue(attr.type)>))
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
RetVal.<attr.name> := <expr>;
<else>
<attr.name> := <expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr
 {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(Def(<scope>).Text)"
tokenLabelPropertyRef_type(scope,attr) ::= "(Def(<scope>).TokenType)"
tokenLabelPropertyRef_line(scope,attr) ::= "(Def(<scope>).Line)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(Def(<scope>).CharPositionInLine)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(Def(<scope>).Channel)"
tokenLabelPropertyRef_index(scope,attr) ::= "(Def(<scope>).TokenIndex)"
```

```
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(StrToIntDef(Def(<scope>).Text,0))"

ruleLabelPropertyRef_start(scope,attr) ::= "(IfThen(Assigned(<scope>), Def(<scope>).Start, nil) as I<labelType>)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(Def(<scope>).Stop as I<labelType>)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(Def(Def(<scope>).Tree as I<ASTLabelType>))"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
IfThen(Assigned(<scope>),
 Input.TokenStream.ToString(
 Input.TreeAdaptor.GetTokenStartIndex(Def(<scope>).Start),
 Input.TreeAdaptor.GetTokenStopIndex(Def(<scope>).Start)), ")
<else>
IfThen(Assigned(<scope>), Input.ToString(
 (Def(<scope>).Start) as IToken,(Def(<scope>).Stop) as IToken), ")
<endif>
>>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> != null) ? <scope>.ST : null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(Def(<scope>).TokenType)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(Def(<scope>).Line)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(IfThen(Assigned(<scope>),Def(<scope>).CharPositionInLine,-1))"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(Def(<scope>).Channel)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(Def(<scope>).TokenIndex)"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(Def(<scope>).Text)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
 "(StrToIntDef(Def(<scope>).Text,0))"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "(RetVal.Start as I<labelType>)"
rulePropertyRef_stop(scope,attr) ::= "(RetVal.Stop as I<labelType>)"
rulePropertyRef_tree(scope,attr) ::= "(RetVal.Tree as I<ASTLabelType>)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
Input.TokenStream.ToString(
 Input.TreeAdaptor.GetTokenStartIndex(RetVal.Start),
 Input.TreeAdaptor.GetTokenStopIndex(RetVal.Start))
<else>
Input.ToString(RetVal.Start as IToken,Input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "RetVal.ST"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "TokenType"
```

```
lexerRulePropertyRef_line(scope,attr) ::= "State.TokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "State.TokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "Channel"
lexerRulePropertyRef_start(scope,attr)
 ::= "State.TokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "StrToInt(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "RetVal.Tree := <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "RetVal.ST := <expr>;"


/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if (<actions.(actionScope).synpredgate>) then
begin
 <action>
end;
<else>
if (State.Backtracking = 0) then
begin
 <action>
end;<\n>
<endif>
<else>
<action>
<endif>
>>


/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
<name> := TBitSet.Create([<words64:{<it>};separator=",">]);<\n>
>>

bitsetDecl(name) ::=
 <<
<name>: IBitSet;<\n>
>>
```

codeFileExtension() ::= ".pas"

true() ::= "True"
false() ::= "False"

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Delphi/Delphi.stg
No license file was found, but licenses were detected in source scan.

/*
New style messages. This file only contains the messages in English, but no
information about which file, line, or column it occurred in.
The location and message ids are taken out of the formats directory.
		Kay Roepke
*/
group en_US;

// TOOL ERRORS

```
// file errors
CANNOT_WRITE_FILE(arg,exception,stackTrace) ::= <<
cannot write file <arg>: <exception>
<stackTrace; separator="\n">
>>
CANNOT_CLOSE_FILE(arg,exception,stackTrace) ::= "cannot close file <arg>: <exception>"
CANNOT_FIND_TOKENS_FILE(arg)
 ::= "cannot find tokens file <arg>"
ERROR_READING_TOKENS_FILE(arg,exception,stackTrace) ::= <<
problem reading token vocabulary file <arg>: <exception>
<stackTrace; separator="\n">
>>
DIR_NOT_FOUND(arg,exception,stackTrace) ::= "directory not found: <arg>"
OUTPUT_DIR_IS_FILE(arg,exception,stackTrace) ::= "output directory is a file: <arg>"
CANNOT_OPEN_FILE(arg,exception,stackTrace) ::= "cannot find or open file: <arg><if(exception)>; reason:
<exception><endif>"
CIRCULAR_DEPENDENCY() ::= "your grammars contain a circular dependency and cannot be sorted into a valid
build order."

INTERNAL_ERROR(arg,arg2,exception,stackTrace) ::= <<
internal error: <arg> <arg2><if(exception)>: <exception><endif>
<stackTrace; separator="\n">
>>
INTERNAL_WARNING(arg) ::= "internal warning: <arg>"
ERROR_CREATING_ARTIFICIAL_RULE(arg,exception,stackTrace) ::= <<
problems creating lexer rule listing all tokens: <exception>
<stackTrace; separator="\n">
>>
TOKENS_FILE_SYNTAX_ERROR(arg,arg2)
 ::=
 "problems parsing token vocabulary file <arg> on line <arg2>"
CANNOT_GEN_DOT_FILE(arg,exception,stackTrace) ::=
 "cannot write DFA DOT file <arg>: <exception>"
BAD_ACTION_AST_STRUCTURE(exception,stackTrace) ::=
 "bad internal tree structure for action '<arg>': <exception>"
BAD_AST_STRUCTURE(arg,exception,stackTrace) ::= <<
bad internal tree structure '<arg>': <exception>
<stackTrace; separator="\n">
>>
FILE_AND_GRAMMAR_NAME_DIFFER(arg,arg2) ::=
 "file <arg2> contains grammar <arg>; names must be identical"
FILENAME_EXTENSION_ERROR(arg) ::=
 "file <arg> must end in a file extension, normally .g"

// code gen errors
MISSING_CODE_GEN_TEMPLATES(arg) ::=
 "cannot find code generation templates <arg>.stg"
MISSING_CYCLIC_DFA_CODE_GEN_TEMPLATES() ::=
 "cannot find code generation cyclic DFA templates for language <arg>"
CODE_GEN_TEMPLATES_INCOMPLETE(arg) ::=
 "at least one code generation template missing for language <arg>"
CANNOT_CREATE_TARGET_GENERATOR(arg,exception,stackTrace)
 ::=
 "cannot create target <arg> code generator: <exception>"
CANNOT_COMPUTE_SAMPLE_INPUT_SEQ() ::=
 "cannot generate a sample input sequence from lookahead DFA"

// grammar interpretation errors
/*
NO_VIABLE_DFA_ALT(arg,arg2) ::=
 "no viable transition from state <arg> on <arg2> while interpreting DFA"
*/

// GRAMMAR ERRORS
SYNTAX_ERROR(arg) ::= "syntax error: <arg>"
RULE_REDEFINITION(arg) ::=
 "rule <arg> redefinition"
LEXER_RULES_NOT_ALLOWED(arg) ::=
 "lexer rule <arg> not allowed in parser"
PARSER_RULES_NOT_ALLOWED(arg) ::=
 "parser rule <arg> not allowed in lexer"
CANNOT_FIND_ATTRIBUTE_NAME_IN_DECL(arg) ::=
 "cannot find an attribute name in attribute declaration"
NO_TOKEN_DEFINITION(arg) ::=
 "no lexer rule corresponding to token: <arg>"
UNDEFINED_RULE_REF(arg) ::=
 "reference to undefined rule: <arg>"
LITERAL_NOT_ASSOCIATED_WITH_LEXER_RULE(arg) ::=
 "literal has no associated lexer rule: <arg>"
CANNOT_ALIAS_TOKENS_IN_LEXER(arg) ::=
 "literals are illegal in lexer tokens{}
 section: <arg>"
ATTRIBUTE_REF_NOT_IN_RULE(arg,arg2) ::=
 "reference to attribute outside of a rule: <arg><if(arg2)>.<arg2><endif>"
UNKNOWN_ATTRIBUTE_IN_SCOPE(arg,arg2) ::=
 "unknown attribute for <arg>: <arg2>"
UNKNOWN_RULE_ATTRIBUTE(arg,arg2) ::=
 "unknown attribute for rule <arg>: <arg2>"
UNKNOWN_SIMPLE_ATTRIBUTE(arg,args2) ::=
 "attribute is not a token, parameter, or return value: <arg>"
ISOLATED_RULE_SCOPE(arg) ::=
 "missing attribute access on rule scope: <arg>"
INVALID_RULE_PARAMETER_REF(arg,arg2) ::=
 "cannot access rule <arg>'s parameter: <arg2>"
INVALID_RULE_SCOPE_ATTRIBUTE_REF(arg,arg2) ::=
 "cannot access rule <arg>'s dynamically-scoped attribute: <arg2>"
```

```
SYMBOL_CONFLICTS_WITH_GLOBAL_SCOPE(arg) ::=
 "symbol <arg> conflicts with global dynamic scope with same name"
WRITE_TO_READONLY_ATTR(arg,arg2,arg3) ::=
 "cannot write to read only attribute: $<arg><if(arg2)>.<arg2><endif>"
LABEL_CONFLICTS_WITH_RULE(arg) ::=
 "label <arg> conflicts with rule with same name"
LABEL_CONFLICTS_WITH_TOKEN(arg)
 ::=
 "label <arg> conflicts with token with same name"
LABEL_CONFLICTS_WITH_RULE_SCOPE_ATTRIBUTE(arg,arg2) ::=
 "label <arg> conflicts with rule <arg2>'s dynamically-scoped attribute with same name"
LABEL_CONFLICTS_WITH_RULE_ARG_RETVAL(arg,arg2) ::=
 "label <arg> conflicts with rule <arg2>'s return value or parameter with same name"
ATTRIBUTE_CONFLICTS_WITH_RULE(arg,arg2) ::=
 "rule <arg2>'s dynamically-scoped attribute <arg> conflicts with the rule name"
ATTRIBUTE_CONFLICTS_WITH_RULE_ARG_RETVAL(arg,arg2) ::=
 "rule <arg2>'s dynamically-scoped attribute <arg> conflicts with<arg2>'s return value or parameter with same
name"
LABEL_TYPE_CONFLICT(arg,arg2) ::=
 "label <arg> type mismatch with previous definition: <arg2>"
ARG_RETVAL_CONFLICT(arg,arg2) ::=
 "rule <arg2>'s argument <arg> conflicts a return value with same name"
NONUNIQUE_REF(arg) ::=
 "<arg> is a non-unique reference"
FORWARD_ELEMENT_REF(arg) ::=
 "illegal forward reference: <arg>"
MISSING_RULE_ARGS(arg)
 ::=
 "missing parameter(s) on rule reference: <arg>"
RULE_HAS_NO_ARGS(arg) ::=
 "rule <arg> has no defined parameters"
ARGS_ON_TOKEN_REF(arg) ::=
 "token reference <arg> may not have parameters"
/*
NONCHAR_RANGE() ::=
 "range operator can only be used in the lexer"
*/
ILLEGAL_OPTION(arg) ::=
 "illegal option <arg>"
LIST_LABEL_INVALID_UNLESS_RETVAL_STRUCT(arg) ::=
 "rule '+=' list labels are not allowed w/o output option: <arg>"
UNDEFINED_TOKEN_REF_IN_REWRITE(arg) ::=
 "reference to undefined token in rewrite rule: <arg>"
REWRITE_ELEMENT_NOT_PRESENT_ON_LHS(arg) ::=
 "reference to rewrite element <arg> without reference on left of ->"
UNDEFINED_LABEL_REF_IN_REWRITE(arg) ::=
 "reference to undefined label in rewrite rule: $<arg>"
NO_GRAMMAR_START_RULE (arg) ::=
 "grammar <arg>: no start rule (no rule can obviously be followed by EOF)"
EMPTY_COMPLEMENT(arg) ::= <<
<if(arg)>
set complement ~<arg> is empty
<else>
set complement is empty
<endif>
>>
UNKNOWN_DYNAMIC_SCOPE(arg) ::=
 "unknown
 dynamic scope: <arg>"
UNKNOWN_DYNAMIC_SCOPE_ATTRIBUTE(arg,arg2) ::=
 "unknown dynamically-scoped attribute for scope <arg>: <arg2>"
RULE_REF_AMBIG_WITH_RULE_IN_ALT(arg) ::=
 "reference $<arg> is ambiguous; rule <arg> is enclosing rule and referenced in the production (assuming enclosing rule)"
ISOLATED_RULE_ATTRIBUTE(arg) ::=
 "reference to locally-defined rule scope attribute without rule name: <arg>"
INVALID_ACTION_SCOPE(arg,arg2) ::=
 "unknown or invalid action scope for <arg2> grammar: <arg>"
ACTION_REDEFINITION(arg) ::=
 "redefinition of <arg> action"
DOUBLE_QUOTES_ILLEGAL(arg) ::=
 "string literals must use single quotes (such as \'begin\'): <arg>"
INVALID_TEMPLATE_ACTION(arg) ::=
 "invalid StringTemplate % shorthand syntax: '<arg>'"
MISSING_ATTRIBUTE_NAME() ::=
 "missing attribute name on $ reference"
ARG_INIT_VALUES_ILLEGAL(arg) ::=
 "rule parameters may not have init values: <arg>"
REWRITE_OR_OP_WITH_NO_OUTPUT_OPTION(arg) ::=
 "<if(arg)>rule <arg> uses <endif>rewrite
 syntax or operator with no output option; setting output=AST"
AST_OP_WITH_NON_AST_OUTPUT_OPTION(arg) ::=
 "AST operator with non-AST output option: <arg>"
NO_RULES(arg) ::= "grammar file <arg> has no rules"
MISSING_AST_TYPE_IN_TREE_GRAMMAR(arg) ::=
 "tree grammar <arg> has no ASTLabelType option"
REWRITE_FOR_MULTI_ELEMENT_ALT(arg) ::=
 "with rewrite=true, alt <arg> not simple node or obvious tree element; text attribute for rule not guaranteed to be correct"
RULE_INVALID_SET(arg) ::=
 "Cannot complement rule <arg>; not a simple set or element"
HETERO_ILLEGAL_IN_REWRITE_ALT(arg) ::=
 "alts with rewrites can't use heterogeneous types left of ->"
NO_SUCH_GRAMMAR_SCOPE(arg,arg2) ::=
 "reference to undefined grammar in rule reference: <arg>.<arg2>"
NO_SUCH_RULE_IN_SCOPE(arg,arg2) ::=
 "rule <arg2> is not defined in grammar <arg>"
TOKEN_ALIAS_CONFLICT(arg,arg2) ::=
 "cannot alias <arg>; string already assigned to <arg2>"
TOKEN_ALIAS_REASSIGNMENT(arg,arg2) ::=
 "cannot alias <arg>;
 token name already assigned to <arg2>"
TOKEN_VOCAB_IN_DELEGATE(arg,arg2) ::=
 "tokenVocab option ignored in imported grammar <arg>"
INVALID_IMPORT(arg,arg2) ::=
 "<arg.grammarTypeString> grammar <arg.name> cannot import <arg2.grammarTypeString> grammar <arg2.name>"
IMPORTED_TOKENS_RULE_EMPTY(arg,arg2) ::=
 "no lexer rules contributed to <arg> from imported grammar <arg2>"
IMPORT_NAME_CLASH(arg,arg2) ::=
 "combined grammar <arg.name> and imported <arg2.grammarTypeString> grammar <arg2.name> both generate <arg2.recognizerName>; import ignored"
AST_OP_IN_ALT_WITH_REWRITE(arg,arg2) ::=
 "rule <arg> alt <arg2> uses rewrite syntax and also an AST operator"
WILDCARD_AS_ROOT(arg) ::= "Wildcard invalid as root; wildcard can itself be a tree"
CONFLICTING_OPTION_IN_TREE_FILTER(arg,arg2) ::= "option <arg>=<arg2> conflicts with tree grammar filter mode"

// GRAMMAR WARNINGS

GRAMMAR_NONDETERMINISM(input,conflictingAlts,paths,disabled,hasPredicateBlockedByAction) ::=
<<
<if(paths)>
Decision
 can match input such as "<input>" using multiple alternatives:
<paths:{  alt <it.alt> via NFA path <it.states; separator=","><\n>}>
<else>
Decision can match input such as "<input>" using multiple alternatives: <conflictingAlts; separator=", ">
<endif>
<if(disabled)><\n>As a result, alternative(s) <disabled; separator=","> were disabled for that input<endif><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by actions.<endif>
>>

DANGLING_STATE(danglingAlts,input) ::= <<
the decision cannot distinguish between alternative(s) <danglingAlts; separator=","> for input such as "<input>"
>>

UNREACHABLE_ALTS(alts) ::= <<
The following alternatives can never be matched: <alts; separator=","><\n>
>>

INSUFFICIENT_PREDICATES(upon,altToLocations,hasPredicateBlockedByAction) ::= <<
Input such as "<upon>" is insufficiently covered with predicates at locations: <altToLocations.keys:{alt|alt <alt>:
```

```
<altToLocations.(alt):{loc| line <loc.line>:<loc.column> at <loc.text>};
 separator=", ">}; separator=", "><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but
were hidden by actions.<endif>
>>

DUPLICATE_SET_ENTRY(arg) ::=
 "duplicate token type <arg> when collapsing subrule into set"

ANALYSIS_ABORTED(enclosingRule) ::= <<
ANTLR could not analyze this decision in rule <enclosingRule>; often this is because of recursive rule references
visible from the left edge of alternatives.  ANTLR will re-analyze the decision with a fixed lookahead of k=1.
Consider using "options {k=1;}" for that decision and possibly adding a syntactic predicate.
>>

RECURSION_OVERLOW(alt,input,targetRules,callSiteStates) ::= <<
Alternative <alt>: after matching input such as <input> decision cannot predict what comes next due to recursion
overflow <targetRules,callSiteStates:{t,c|to <t> from <c:{s|<s.enclosingRule.name>};separator=", ">}; separator="
and ">
>>

LEFT_RECURSION(targetRules,alt,callSiteStates) ::= <<
Alternative <alt> discovers infinite left-recursion
 <targetRules,callSiteStates:{t,c|to <t> from <c:{s|<s.enclosingRule>};separator=", ">}; separator=" and ">
>>

UNREACHABLE_TOKENS(tokens) ::= <<
The following token definitions can never be matched because prior tokens match the same input: <tokens;
separator=",">
>>

TOKEN_NONDETERMINISM(input,conflictingTokens,paths,disabled,hasPredicateBlockedByAction) ::=
<<
<if(paths)>
Decision can match input such as "<input>" using multiple alternatives:
<paths:{  alt <it.alt> via NFA path <it.states; separator=","><\n>}>
<else>
Multiple token rules can match input such as "<input>": <conflictingTokens; separator=", "><\n>
<endif>
<if(disabled)><\n>As a result, token(s) <disabled; separator=","> were disabled for that
input<endif><if(hasPredicateBlockedByAction)><\n>Semantic predicates were present but were hidden by
actions.<endif>
>>

LEFT_RECURSION_CYCLES(listOfCycles) ::= <<
The following sets of rules are mutually left-recursive <listOfCycles:{c| [<c:{r|<r.name>}; separator=", ">]};
 separator=" and ">
>>
```

```
NONREGULAR_DECISION(ruleName,alts) ::= <<
[fatal] rule <ruleName> has non-LL(*) decision due to recursive rule invocations reachable from alts <alts;
separator=",">.  Resolve by left-factoring or using syntactic predicates or using backtrack=true option.
>>

/* l10n for message levels */
warning() ::= "warning"
error() ::= "error"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/templates/messages/languages/en.stg
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : C# by code generator.
*/
```

```
group ASTDbg;

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor;
public ITreeAdaptor TreeAdaptor
{
 get {
<if(grammar.grammarIsRoot)>
  return this.adaptor;
<else>
  this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
      <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
 set { this.adaptor = new DebugTreeAdaptor(dbg, value); }
}<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy dbg = new DebugEventSocketProxy(this, port, adaptor);
DebugListener = dbg;
<!
Original line follows, replaced by the next two ifs:
set<inputStreamType>(new Debug<inputStreamType>(input,dbg));
!>
<if(PARSER)>
TokenStream = new DebugTokenStream(input,dbg);<\n>
<endif>
<if(TREE_PARSER)>
TokenStream = new DebugTreeNodeStream(input,dbg);<\n>
<endif>
try {
   dbg.Handshake();
} catch (IOException ioe) {
   ReportError(ioe);
}
>>

@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
dbg.TreeAdaptor = adap;
>>
```

```
@ctorForProfilingRootGrammar.finally() ::=<<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
dbg.TreeAdaptor = adap;
>>

@ctorForPredefinedListener.superClassRef()
 ::= "base(input, dbg)"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>

@rewriteElement.pregen() ::= "dbg.Location(<e.line>,<e.pos>);"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/PythonTarget.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/RubyTarget.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2008 Erik van Bilsen
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES,
 INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Templates for building ASTs during tree parsing.
*
* Deal with many combinations. Dimensions are:
* Auto build or rewrite
*   no label, label, list label (label/no-label handled together)
*   child, root

```
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
 */
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_First[0] := nil;
_Last := nil;<\n>
>>

ruleDeclarationVars() ::= <<
<super.ruleDeclarationVars()>
_First, _Save_Last: array [0..63] of I<ASTLabelType>;
_Last: I<ASTLabelType>;
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if (State.Backtracking = 0) then
begin<endif>
<if(rewriteMode)>
RetVal.Tree := _First[0] as I<ASTLabelType>;
if (Adaptor.GetParent(RetVal.Tree) \<\> nil) and (Adaptor.IsNil(Adaptor.GetParent(RetVal.Tree))) then
 RetVal.Tree := Adaptor.GetParent(RetVal.Tree) as I<ASTLabelType>;
<endif>
<if(backtracking)>end;<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_Last := Input.LT(1)
 as I<ASTLabelType>;
begin
 _Save_Last[<treeLevel>] := _Last;
 _First[<treeLevel>] := nil;
<if(!rewriteMode)>
 Root[<treeLevel>] := Adaptor.GetNilNode as I<ASTLabelType>;<\n>
<endif>
 <root:element()>
```

```
<if(rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
<if(root.el.rule)>
 if (_First[<enclosingTreeLevel>] = nil) then _First[<enclosingTreeLevel>] := <root.el.label>.Tree;
<else>
 if (_First[<enclosingTreeLevel>] = nil) then _First[<enclosingTreeLevel>] := <root.el.label>;
<endif>
<endif>
 <actionsAfterRoot:element()>
<if(nullableChildList)>
 if (Input.LA(1) = TToken.DOWN) then
 begin
   Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
 end;
<else>
 Match(Input, TToken.DOWN, nil); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(Input, TToken.UP, nil); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
 Adaptor.AddChild(Root[<enclosingTreeLevel>],
 Root[<treeLevel>]);
<endif>
 _Last := _Save_Last[<treeLevel>];
end;<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
```

```
<if(hetero)>
<label>_tree := T<hetero>.Create(<label>);
<else>
 <label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
 Adaptor.AddChild(Root[<treeLevel>], <label>_tree);
<if(backtracking)>
end;
<endif>
<else> <! rewrite mode !>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
if (_First[<treeLevel>] = nil) then _First[<treeLevel>] := <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<if(hetero)>
<label>_tree := T<hetero>.Create(<label>);
<else>
 <label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
 Root[<treeLevel>] := Adaptor.BecomeRoot(<label>_tree, Root[<treeLevel>]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>
```

```
// SET AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_Llast := Input.LT(1) as I<ASTLabelType>;
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then begin <endif>
<if(hetero)>
<label>_tree
 := T<hetero>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Adaptor.AddChild(Root[<treeLevel>], <label>_tree);
<if(backtracking)>end;<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then begin <endif>
<if(hetero)>
<label>_tree := T<hetero>.Create(<label>);
<else>
<label>_tree := Adaptor.DupNode(<label>) as I<ASTLabelType>;
<endif><\n>
Root[<treeLevel>] := Adaptor.BecomeRoot(<label>_tree, Root[<treeLevel>]) as I<ASTLabelType>;
<if(backtracking)>end;<endif>
<endif>
}
)>
>>

// RULE REF AST
```

```
/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope)
 ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>
<if(!rewriteMode)>
Adaptor.AddChild(Root[<treeLevel>], <label>.Tree);
<else> <! rewrite mode !>
if (_First[<treeLevel>] = nil) then _First[<treeLevel>] := <label>.Tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[<treeLevel>] :=
Adaptor.BecomeRoot(<label>.Tree, Root[<treeLevel>]) as I<ASTLabelType>;
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

/**
 rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefTrackAndListLabel(...)>
>>
```

```
/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_Last := Input.LT(1) as I<ASTLabelType>;
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
T<hetero>.Create((Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode)
<else>
(Locals['Stream_<token>']
 as IRewriteRuleElementStream).NextNode
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
RetVal.Tree := Adaptor.RulePostProcessing(Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Delphi/ASTTreeParser.stg

No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

finishedBacktracking(block) ::= <<
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
    <block>
<else>
<block>
<endif>
>>

/** Add a variable to track last element matched */
```

```
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
_first_0 = None
_last = None<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<finishedBacktracking({
<if(rewriteMode)>
retval.tree = _first_0
if self._adaptor.getParent(retval.tree) is not None and self._adaptor.isNil(self._adaptor.getParent(retval.tree)):
   retval.tree = self._adaptor.getParent(retval.tree)
<endif>
})>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = self.input.LT(1)
_save_last_<treeLevel> = _last
_first_<treeLevel> = None
<if(!rewriteMode)>
root_<treeLevel> = self._adaptor.nil()<\n>
<endif>
<root:element()>
<if(rewriteMode)>
<finishedBacktracking({
<if(root.el.rule)>
if
 _first_<enclosingTreeLevel> is None:
   _first_<enclosingTreeLevel> = <root.el.label>.tree<\n>
<else>
if _first_<enclosingTreeLevel> is None:
   _first_<enclosingTreeLevel> = <root.el.label><\n>
<endif>
})>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if self.input.LA(1) == DOWN:
   self.match(self.input, DOWN, None)
   <children:element()>
   self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)<\n>
<endif>
<if(!rewriteMode)>
self._adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>)<\n>
<endif>
_last = _save_last_<treeLevel>

>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(hetero)>
<label>_tree
 = <hetero>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label><\n>
})>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
```

```
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = self.input.LT(1)
<super.tokenRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<if(hetero)>
<label>_tree = <hetero>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/**
 Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,hetero) ::= <<
_last = self.input.LT(1)
<super.wildcard(...)>
<if(!rewriteMode)>
<finishedBacktracking({
<label>_tree = self._adaptor.dupTree(<label>)
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<else> <! rewrite mode !>
<finishedBacktracking({
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label>
})>
<endif>
>>

// SET AST
matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = self.input.LT(1)
<super.matchSet(..., postmatchCode={
```

```
<if(!rewriteMode)>
<finishedBacktracking({
<if(hetero)>
<label>_tree = <hetero>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
self._adaptor.addChild(root_<treeLevel>, <label>_tree)
})>
<endif>
})>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last
 = self.input.LT(1)
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<finishedBacktracking({
<if(hetero)>
<label>_tree = <hetero>(<label>)
<else>
<label>_tree = self._adaptor.dupNode(<label>)
<endif><\n>
root_<treeLevel> = self._adaptor.becomeRoot(<label>_tree, root_<treeLevel>)
})>
<endif>
})>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRef(...)>
<finishedBacktracking({
<if(!rewriteMode)>
self._adaptor.addChild(root_<treeLevel>, <label>.tree)
```

```
<else> <! rewrite mode !>
if _first_<treeLevel> is None:
   _first_<treeLevel> = <label>.tree<\n>
<endif>
})>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last
 = self.input.LT(1)
<super.ruleRef(...)>
<if(!rewriteMode)>
<finishedBacktracking({
root_<treeLevel> = self._adaptor.becomeRoot(<label>.tree, root_<treeLevel>)
})>
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".tree",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefRootTrack(...)>
>>
```

```
/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = self.input.LT(1)
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/**
 Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
<hetero>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
})>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add
 a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _last;<\n>
<ASTLabelType> _first_0;<\n>
>>

/** Add a variable to track last element matched */
ruleInitializations() ::= <<
<super.ruleInitializations()>
_last = NULL;<\n>
_first_0 = NULL;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
```

```
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( ADAPTOR->getParent(ADAPTOR, retval.tree) != NULL && ADAPTOR->isNilNode(ADAPTOR,
ADAPTOR->getParent(ADAPTOR, retval.tree) ) )
{
   retval.tree = (<ASTLabelType>)ADAPTOR->getParent(ADAPTOR, retval.tree);
}
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel)
 ::= <<
_last = (<ASTLabelType>)LT(1);
{
<ASTLabelType> _save_last_<treeLevel>;
<ASTLabelType> _first_last_<treeLevel>;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel>;
<endif>
_save_last_<treeLevel> = _last;
_first_last_<treeLevel> = NULL;
<if(!rewriteMode)>
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel> == NULL ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel> == NULL ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<if(backtracking)>}<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( LA(1)==ANTLR3_TOKEN_DOWN ) {
   MATCHT(ANTLR3_TOKEN_DOWN, NULL);
   <children:element()>
   MATCHT(ANTLR3_TOKEN_UP, NULL);
}
<else>
MATCHT(ANTLR3_TOKEN_DOWN, NULL);
<children:element()>
MATCHT(ANTLR3_TOKEN_UP, NULL);
<endif>
<if(!rewriteMode)>
ADAPTOR->addChild(ADAPTOR,
 root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
<if(hetero)>
<label>_tree = <hetero>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else>
<if(backtracking)>if ( BACKTRACKING ==0 ) {<endif>
if ( _first_<treeLevel> == NULL ) _first_<treeLevel> = <label>;
<if(backtracking)>}<endif>
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>
```

```
/**
 ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(hetero)>
<label>_tree = <hetero>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(hetero)>
<label>_tree = <hetero>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR,
 <label>);
<endif><\n>
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>
```

```
matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) {<endif>
<if(hetero)>
<label>_tree = <hetero>New(<label>);
<else>
<label>_tree = (<ASTLabelType>)ADAPTOR->dupNode(ADAPTOR, <label>);
<endif>
root_<treeLevel> = (<ASTLabelType>)ADAPTOR->becomeRoot(ADAPTOR, <label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRef(...)>
<if(backtracking)>if
 ( BACKTRACKING == 0 )
{
<endif>
<if(!rewriteMode)>
 ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <label>.tree);
<else>
 if ( _first_<treeLevel> == NULL ) _first_<treeLevel> = <label>.tree;
<endif>
<if(backtracking)>}<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<super.listLabelAST(elem=label,...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
```

```
_last = (<ASTLabelType>)LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( BACKTRACKING == 0 ) <endif>root_<treeLevel> = (<ASTLabelType>)(ADAPTOR-
>becomeRoot(ADAPTOR, <label>.tree, root_<treeLevel>));
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<super.listLabelAST(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefTrack(...)>
>>

/**
 x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefTrackAndListLabel(...)>
>>
/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>



/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
<hetero>New(stream_<token>->nextNode(stream_<token>))
<else>
stream_<token>->nextNode(stream_<token>)
<endif>
```

```
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp(...)>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<if(!ruleDescriptor.isSynPred)>
 retval.stop
 = LT(-1);<\n>
<endif>
 retval.tree = (<ASTLabelType>)ADAPTOR->rulePostProcessing(ADAPTOR, root_0);
<if(backtracking)>
}
<endif>
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);}; separator="\n">
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/C/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2006 Kunle Odutola
Copyright (c) 2005 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS
 FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/CSharpTarget.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/CSharp2Target.java
No license file was found, but licenses were detected in source scan.

/** Template overrides to add debugging to normal Java output;

```
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
group Dbg;

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.Runtime.Debug;
using IOException = System.IO.IOException;
>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
public
 static readonly string[] ruleNames =
 new string[]
 {
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n ", separator=", ">
 };<\n>
<endif>
<if(grammar.grammarIsRoot)><! grammar imports other grammar(s) !>
 int ruleLevel = 0;
 public virtual int RuleLevel { get { return ruleLevel; } }
 public virtual void IncRuleLevel() { ruleLevel++; }
 public virtual void DecRuleLevel() { ruleLevel--; }
<if(profile)>
 <ctorForProfilingRootGrammar()>
<else>
 <ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else><! imported grammar !>
 public int RuleLevel { get { return <grammar.delegators:{g| <g:delegateName()>}>.RuleLevel; } }
 public void IncRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.IncRuleLevel(); }
 public void DecRuleLevel() { <grammar.delegators:{g| <g:delegateName()>}>.DecRuleLevel(); }
 <ctorForDelegateGrammar()>
<endif>
<if(profile)>
public virtual bool AlreadyParsedRule( IIntStream input, int ruleIndex )
{
 ((Profiler)dbg).ExamineRuleMemoization(input,
 ruleIndex, <grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
 return super.AlreadyParsedRule(input, ruleIndex);
}<\n>
public virtual void Memoize( IIntStream input, int ruleIndex, int ruleStartIndex )
{
 ((Profiler)dbg).Memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
 super.Memoize(input, ruleIndex, ruleStartIndex);
}<\n>
<endif>
protected virtual bool EvalPredicate( bool result, string predicate )
{
 dbg.SemanticPredicate( result, predicate );
 return result;
}<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we add port number and profile stuff if root grammar !>
public <name>( <inputStreamType> input )
 : this( input, DebugEventSocketProxy.DEFAULT_DEBUGGER_PORT, new RecognizerSharedState() )
{
}
public <name>( <inputStreamType> input, int port, RecognizerSharedState state )
 : base( input, state )
{
 <parserCtorBody()>
 <createListenerAndHandshake()>
 <grammar.directDelegates:{g|<g:delegateName()>
 = new <g.recognizerName>( input, dbg, this.state, this<grammar.delegators:{g|, <g:delegateName()>}> );};
separator="\n">
 <@finally()>
}<\n>
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
public <name>( <inputStreamType> input )
 : this( input, new Profiler(null), new RecognizerSharedState() )
{
}
public <name>( <inputStreamType> input, IDebugEventListener dbg, RecognizerSharedState state )
 : base( input, dbg, state )
{
 Profiler p = (Profiler)dbg;
 p.setParser(this);
 <parserCtorBody()>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, dbg, this.state, this<grammar.delegators:{g|,
<g:delegateName()>}> );}; separator="\n">
 <@finally()>
}
<\n>
>>
```

```
/** Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
public <name>( <inputStreamType> input, IDebugEventListener dbg, RecognizerSharedState
 state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}> )
 : base( input, dbg, state )
{
 <parserCtorBody()>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>( input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}> );}; separator="\n">
}<\n>
>>

ctorForPredefinedListener() ::= <<
public <name>( <inputStreamType> input, IDebugEventListener dbg )
 <@superClassRef>: base( input, dbg, new RecognizerSharedState() )<@end>
{
<if(profile)>
 Profiler p = (Profiler)dbg;
 p.setParser(this);
<endif>
 <parserCtorBody()>
 <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, dbg, this.state,
this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
 <@finally()>
}<\n>
>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port, input.TreeAdaptor );<\n>
<else>
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port,
 null );<\n>
<endif>
DebugListener = proxy;
try
{
 proxy.Handshake();
}
catch ( IOException ioe )
{
 ReportError( ioe );
}
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"
```

```
@rule.preamble() ::= <<
try
{
 dbg.EnterRule( GrammarFileName, "<ruleName>" );
 if ( RuleLevel == 0 )
 {
  dbg.Commence();
 }
 IncRuleLevel();
 dbg.Location( <ruleDescriptor.tree.line>, <ruleDescriptor.tree.charPositionInLine> );<\n>
>>

@rule.postamble() ::= <<
dbg.Location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.charPositionInLine>);<\n>
}
finally
{
 dbg.ExitRule( GrammarFileName, "<ruleName>" );
 DecRuleLevel();
 if ( RuleLevel == 0 )
 {
  dbg.Terminate();
 }
}<\n>
>>

@synpred.start() ::= "dbg.BeginBacktrack( state.backtracking );"

@synpred.stop() ::= "dbg.EndBacktrack( state.backtracking, success );"

// Common debug event triggers used by region overrides below

enterSubRule() ::= <<
try
{
 dbg.EnterSubRule( <decisionNumber> );<\n>
>>

exitSubRule() ::= <<
}
finally
{
 dbg.ExitSubRule(
 <decisionNumber> );
}<\n>
>>

enterDecision() ::= <<
```

```
try
{
 dbg.EnterDecision( <decisionNumber> );<\n>
>>

exitDecision() ::= <<
}
finally
{
 dbg.ExitDecision( <decisionNumber> );
}<\n>
>>

enterAlt(n) ::= "dbg.EnterAlt( <n> );<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
 "dbg.RecognitionException( eee<decisionNumber>
 );<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"
```

```
@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
 "dbg.Location( <it.line>, <it.pos> );"

@matchSet.mismatchedSetException() ::=
 "dbg.RecognitionException( mse );"

@dfaState.noViableAltException() ::= "dbg.RecognitionException( nvae );"

@dfaStateSwitch.noViableAltException() ::= "dbg.RecognitionException( nvae );"

dfaDecision(decisionNumber,description) ::= <<
try
{
 isCyclicDecision = true;
 <super.dfaDecision(...)>
}
catch ( NoViableAltException nvae )
{
 dbg.RecognitionException( nvae );
 throw nvae;
}
>>

@cyclicDFA.errorMethod() ::= <<
public override void Error( NoViableAltException nvae )
{
 ((DebugParser)recognizer).dbg.RecognitionException( nvae );
}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description)
 ::= <<
EvalPredicate( <pred>, "<description>" )
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/Dbg.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]

Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
group Cpp implements ANTLRCore;

cppTypeInitMap ::= [
 "int":"0",
 "long":"0",
 "float":"0.0",
 "double":"0.0",
 "bool":"false",
 "byte":"0",
 "short":"0",
 "char":"0",
 default:"0" // anything other than an atomic type
]

// What we generate lexer/parser/treeparser, used a suffix in a few places
generatedType() ::= <<
<if(LEXER)>Lexer<endif><if(PARSER)>Parser<endif><if(TREE_PARSER)>TreeParser<endif>
>>

leadIn(type) ::=
<<
/** \file
```

```
*
*  This <type> file was generated by ANTLR
 version <ANTLRVersion>
*
*     -  From the grammar source file : <fileName>
*     -                            On : <generatedTimestamp>
<if(LEXER)>
*     -                 for the lexer : <name><\n>
<endif>
<if(PARSER)>
*     -                for the parser : <name><\n>
<endif>
<if(TREE_PARSER)>
*     -           for the tree parser : <name><\n>
<endif>
*
*  Edit at your own peril.
*/
>>

standardHeaders() ::=
<<
#include \<antlr3/<generatedType()>.h>

<if(profile)>
#warning "No profiling support.."
<endif>
<if(TREE_PARSER)>
#warning "No tree parsing yet..."
<endif>
>>

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
     docComment, recognizer,
     name, tokens, tokenNames, rules, cyclicDFAs,
     bitsets, buildTemplate, profile,
     backtracking, synpreds, memoize, numRules,
     fileName, ANTLRVersion, generatedTimestamp, trace,
     scopes, superClass)
 ::=
<<
<leadIn("C++ source")>
<@includes>
#include "<name><headerFileExtension()>"
<@end>
```

```
<if(actions.(actionScope).header)>
// Header action start ========================================================
<actions.(actionScope).header>
// Header action end   ========================================================
<endif>

<headerAction>

<standardHeaders()>

<docComment>
<recognizer>
>>
parserHeaderFile() ::= <<
>>
treeParserHeaderFile() ::= <<
>>
lexerHeaderFile() ::= <<
template\<typename StreamType, typename TokenType, typename TokenBuilder>
class <name> : public antlr3::Lexer\<StreamType,TokenType,TokenBuilder> {
 // carry over general types
 typedef typename StreamType::position_type position_type;
 typedef typename StreamType::char_type char_type;

 typedef antlr3::tokenid_type              tokenid_type;
 typedef antlr3::channel_type              channel_type;
 typedef antlr3::decision_type             decision_type;
 // exception shorthands
 typedef antlr3::MismatchException\<position_type,char_type>
        MismatchException;
 typedef antlr3::MismatchedRangeException\<position_type,char_type> MismatchedRangeException;
 typedef antlr3::MismatchedSetException\<position_type,char_type>   MismatchedSetException;
 typedef antlr3::EarlyExitException\<position_type>       EarlyExitException;
 typedef antlr3::NoViableAltException\<position_type>     NoViableAltException;
<if(backtracking)>
 // @TODO backtracking ruleMemo = new HashMap[<numRules>+1];<\n> <! index from 1..n !>
<endif>

public:
 <tokens:{static const tokenid_type <tokenPrefix()><it.name> = <it.type>;}; separator="\n">
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <actions.lexer.members>

 <name>(StreamType* input)
 : antlr3::Lexer\<StreamType,TokenType,TokenBuilder>(input)
 {
 }
<!if(filterMode)!>
   <!filteringNextToken()!>
<!endif!>
 <rules; separator="\n\n">

 // syn preds
 <synpreds:{p | <lexerSynpred(p)>}>

 // cyclic dfa's
 <cyclicDFAs:{dfa | DFA<dfa.decisionNumber>
 dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>(this);}; separator="\n">
 // dfa tables..
}; // class <name><\n>
>>

headerFile( LEXER,
          PARSER,
          TREE_PARSER,
          actionScope,
          actions,
          docComment,
          recognizer,
          name,
          tokens,
          tokenNames,
          rules,
          cyclicDFAs,
          bitsets,
          buildTemplate,
          profile,
          backtracking,
          synpreds,
          memoize,
          numRules,
          fileName,
          ANTLRVersion,
          generatedTimestamp,
          trace,
          scopes,
          superClass
      ) ::=
<<
#ifndef _<name>_H
#define _<name>_H
<leadIn("C++ header")>
<actions.(actionScope).headerfile>

<@includes>
```

```
<standardHeaders()>
<@end>

<if(LEXER)>
<lexerHeaderFile()>
<endif>
<if(PARSER)>
<parserHeaderFile()>
<endif>
<if(TREE_PARSER)>
<treeParserHeaderFile()>
<endif>


#endif //
 _<name>_H<\n>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode) ::= <<

<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

>>

filteringNextToken() ::= <<
/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
public Token nextToken() {
 while (true) {
  if ( input.LA(1)==CharStream.EOF ) {
   return Token.EOF_TOKEN;
  }
  this->token = 0;
  tokenStartCharIndex = getCharIndex();
  try {
   int m = input.mark();
   backtracking=1; <! means we won't throw slow exception !>
   failed=false;
   mTokens();
   backtracking=0;
   <! mTokens backtracks with synpred at backtracking==2
    and we set the synpredgate to allow actions at level 1. !>
   if ( failed ) {
    input.rewind(m);
    input.consume();
 <! advance one char and try again !>
   }
   else {
    return token;
   }
  }
  catch (RecognitionException re) {
   // shouldn't happen in backtracking mode, but...
   reportError(re);
   recover(re);
  }
 }
}

public void memoize(IntStream input, int ruleIndex, int ruleStartIndex)
{
 if ( backtracking > 1 )
  super.memoize(input, ruleIndex, ruleStartIndex);
}

public boolean alreadyParsedRule(IntStream input, int ruleIndex)
{
 if ( backtracking > 1 )
  return super.alreadyParsedRule(input, ruleIndex);
 return false;
}
>>

filteringActionGate() ::= "backtracking == 1"

/** How to generate a parser */
genericParser(
 grammar, name, scopes, tokens, tokenNames, rules, numRules, cyclicDFAs,
 bitsets, inputStreamType, superClass, ASTLabelType="Object",
 labelType, members, filterMode
 ) ::= <<
// genericParser
class <name> : public <@superClassName><superClass><@end> {
public:
 static const char* tokenNames[] = {
  "\<invalid>", "\<EOR>", "\<DOWN>",
 "\<UP>", <tokenNames; separator=", ">
 };
 <tokens:{static tokenid_type <tokenPrefix()><it.name>=<it.type>;}; separator="\n">
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <@members>
 <name>(StreamType* input)
 : <superClass>\<StreamType,TokenType>(input)
 {
<if(backtracking)>
  ruleMemo = new HashMap[<numRules>+1];<\n> <! index from 1..n !>
<endif>
 }
 <@end>

 //@TODO public String[] getTokenNames() { return tokenNames; }
 //@TODO public String getGrammarFileName() { return "<fileName>"; }
 <members>

 <rules; separator="\n\n">

 <synpreds:{p | <synpred(p)>}>

 <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
 <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

 <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
       words64=it.bits)>
};
>>

parser(
 grammar, name, scopes, tokens, tokenNames,
 rules, numRules, bitsets, ASTLabelType,

 superClass="Parser", labelType="Token",
 members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="TokenStream", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction,
 rules, numRules,
 bitsets,
 labelType={<ASTLabelType>}, ASTLabelType="Object",
 superClass="TreeParser", members={<actions.treeparser.members>}, filterMode
 ) ::= <<
<genericParser(inputStreamType="TreeNodeStream", ...)>
>>
```

```
/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description,
 nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) throws
RecognitionException {
 <if(trace)>System.out.println("enter <ruleName> "+input.LT(1)+" failed="+failed+"
backtracking="+backtracking);<endif>
<if(trace)>
 try {
  <block>
 }
 finally {
  System.out.println("exit <ruleName> "+input.LT(1)+" failed="+failed+" backtracking="+backtracking);
 }
<else>
 <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public boolean <name>() {
   this->backtracking++;
   <@start()>
   int start = input.mark();
   try {
       <name>_fragment(); // can never throw exception
   } catch (RecognitionException re) {
       System.err.println("impossible: "+re);
   }
   boolean success = ! this->failed;
   input.rewind(start);
   <@stop()>
   this->backtracking--;
   this->failed = false;
   return success;
}<\n>
>>
```

```
lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name)
 ::= <<
<if(memoize)>
if ( backtracking > 0 && alreadyParsedRule(input, <ruleDescriptor.index>) )
 return <ruleReturnValue()>;
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>
if (failed)
 return <ruleReturnValue()>;
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if (backtracking > 0)
{
 failed = true;
 return <ruleReturnValue()>;
}
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start <ruleName>
// <fileName>:<description>
public <returnType()> <ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
throw(antlr3::BaseRecognitionException)

{
<if(trace)>
 antlr3::Tracer trace(this,"<ruleName>");
 System.out.println("enter <ruleName> "+input.LT(1)+" failed="+failed+" backtracking="+backtracking);
<endif>
```

```
 <ruleDeclarations()>
 <ruleLabelDefs()>
 <ruleDescriptor.actions.init>
 <@preamble()>
 try {
  <ruleMemoization(name=ruleName)>
  <block>
 }
<if(exceptions)>
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
 <actions.(actionScope).rulecatch>
<else>
 catch (RecognitionException re) {
  reportError(re);
  recover(input,re);
 }<\n>
<endif>
<endif>
<endif>
 finally {
  <if(trace)>System.out.println("exit <ruleName> "+input.LT(1)+" failed="+failed+"
backtracking="+backtracking);<endif>
  <ruleCleanUp()>
  <(ruleDescriptor.actions.finally):execAction()>
 }
 <@postamble()>
 return <ruleReturnValue()>;
}
// $ANTLR end <ruleName>
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
   <e.action>
}
>>

ruleDeclarations() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.push(new
 <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.push(new <it.name>_scope());}; separator="\n">
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = new <returnType()>();
retval.start = input.LT(1);<\n>
<else>
```

```
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.index();
<endif>
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
  :{<labelType> <it.label.text>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
  :{List list_<it.label.text>=null;}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
  :ruleLabelDef(label=it); separator="\n"
>
<[ruleDescriptor.allRuleRefsInAltsWithRewrites,ruleDescriptor.allTokenRefsInAltsWithRewrites]
  :{List list_<it>=new ArrayList();}; separator="\n"
>
>>

ruleReturnValue()
 ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.pop();}; separator="\n">
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.stop = input.LT(-1);<\n>
<endif>
<if(memoize)>
<if(backtracking)>
if ( backtracking > 0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
```

```
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
void m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
throw(antlr3::BaseRecognitionException)
{
<if(trace)>
 antlr3::Tracer trace(this,"<ruleName>");
<endif>
 antlr3::CountScope nestingTracker(this->ruleNestingLevel);
 StreamType&
 input(this->getInput());
<if(nakedBlock)>
 <ruleDescriptor.actions.init>
 <ruleMemoization(name=ruleName)>
 <block><\n>
<else>
 tokenid_type type = <tokenPrefix()><ruleName>;
 channel_type channel = antlr3::Token::DEFAULT_CHANNEL;
 position_type start(input.getPosition());
 <ruleDescriptor.actions.init>
 <ruleMemoization(name=ruleName)>
 <block>
 <! create token if none exists *and* we are an outermost token rule !>
 <execAction({if ( this->token == 0 && this->ruleNestingLevel == 1 ) {
 TokenType *tt = TokenBuilder::build(type,start,input,channel);
 std::cout \<\< (*tt) \<\< std::endl;
 this->emit(tt);
 }<\n>
})>
<endif>
}
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
void mTokens() throw(antlr3::BaseRecognitionException)
{
 StreamType& input(this->getInput());
 <block><\n>
}
>>
```

```
// S U B R U L E S

/** A (...) subrule
 with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,
 maxK,maxAlt,description) ::= <<
// block <fileName>:<description>
decision_type alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// ruleBlock <fileName>:<description>
decision_type alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// ruleBlockSingleAlt <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description)
 ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
```

```
>>

/** A (..)+ block with 0 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// positiveClosureBlock <fileName>:<description>
decision_type cnt<decisionNumber>=0;
<decls>
<@preloop()>
do {
 decision_type alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
  if ( cnt<decisionNumber> >= 1 )
   goto loop<decisionNumber>;
          EarlyExitException eee( input.getPosition(), <decisionNumber> );
    <@earlyExitException()>
          throw eee;
 }
 cnt<decisionNumber>++;
} while (true);
loop<decisionNumber>: ;
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<
// closureBlock <fileName>:<description>
<decls>
<@preloop()>
do {
 decision_type alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
  goto loop<decisionNumber>;
 }
} while (true);
```

```
loop<decisionNumber>: ;
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> :
 <@prealt()>
 <it>
 break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt)
 ::= <<
// alt <fileName>:<description>
{
 <@declarations()>
 <elements:element()>
 <@cleanup()>
}
>>

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
// element <fileName>:<description>
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex) ::= <<
// tokenRef
<if(label)>
```

```
<label> = input.LT(1);<\n>
<endif>
this->match(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID no AST building */
tokenRefAndListLabel(token,label,elementIndex) ::= <<
<tokenRef(...)>
<listLabel(...)>
>>

listLabel(label) ::= <<
if (list_<label>==null) list_<label>=new ArrayList();
list_<label>.add(<label>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
// charRef
<if(label)>
<tokenid_type()> <label> = input.LA(1);<\n>
<endif>
this->match(<char>);
<checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b)
 ::= "this->matchRange(<a>,<b>); <checkRuleBacktrackFailure()>"

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
// matchSet
<if(label)>
<label> = input.LT(1);<\n>
<endif>
if ( <s> )
{
 <postmatchCode>
 input.consume();
<if(!LEXER)>
 errorRecovery=false;
<endif>
 <if(backtracking)>failed=false;<endif>
}
else
{
 <ruleBacktrackFailure()>
 MismatchedSetException mse(input.getPosition(),input.LA(1));
 <@mismatchedSetException()>
<if(LEXER)>
 this->recover(mse);
<else>
 this->recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
<endif>
 throw mse;
}<\n>
>>

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
// lexerStringRef
<if(label)>
position_type <label>Start(input.getPosition());
this->match( <string> );
<checkRuleBacktrackFailure()>
TokenType* <label> =
TokenBuilder::build(Token.INVALID_TOKEN_TYPE,<label>Start,input,Token.DEFAULT_CHANNEL);
<else>
this->match(
 <string> );
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label> = input.LT(1);<\n>
<endif>
this->matchAny( input );
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(...)>
>>

/** Match . wildcard */
wildcardChar(label, elementIndex) ::= <<
```

```
<if(label)>
<tokenid_type()> <label> = input.LA(1);<\n>
<endif>
this->matchAny();
<checkRuleBacktrackFailure()>
>>

tokenid_type() ::= "<if(LEXER)>char_type<else>tokenid_type<endif>"

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.
*/
ruleRef(rule,label,elementIndex,args) ::= <<
following.push(FOLLOW_<rule>_in_<ruleName><elementIndex>);
<if(label)>
<label>=<rule>(<args>);<\n>
<else>
<rule>(<args>);<\n>
<endif>
following.pop();
<checkRuleBacktrackFailure()>
>>

/**
 ids+=ID */
ruleRefAndListLabel(rule,label,elementIndex,args) ::= <<
<ruleRef(...)>
<listLabel(...)>
>>

/** A lexer rule reference */
lexerRuleRef(rule,label,args) ::= <<
<if(label)>
position_type <label>Start(input.getPosition());
m<rule>(<args>);
<checkRuleBacktrackFailure()>
TokenType* <label> =
TokenBuilder::build(Token.INVALID_TOKEN_TYPE,<label>Start,input,Token.DEFAULT_CHANNEL);
<else>
m<rule>(<args>);
<checkRuleBacktrackFailure()>
<endif>
>>
```

```
/** EOF in the lexer */
lexerMatchEOF(label) ::= <<
<if(label)>
position_type <label>Start(input.getPosition());
match(EOF);
<checkRuleBacktrackFailure()>
TokenType* <label> = TokenBuilder::build(Token.EOF,<label>Start,input,Token.DEFAULT_CHANNEL);
<else>
match(EOF);
<checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, children, nullableChildList) ::= <<
<root:element()>
<if(nullableChildList)>
if ( input.LA(1)==antlr3::Token::DOWN ) {
    match(input, antlr3::Token::DOWN, null);
   <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, antlr3::Token::UP, null);
   <checkRuleBacktrackFailure()>
}
<else>
match(input, antlr3::Token::DOWN, null);
<checkRuleBacktrackFailure()>
<children:element()>
match(input, antlr3::Token::UP, null);
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
 <ruleBacktrackFailure()>
 throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)
dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<if(!semPredState)>
<tokenid_type()> LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
```

```
<endif>
<edges; separator="\nelse ">
else
{
<if(eotPredictsAlt)>
 alt<decisionNumber> = <eotPredictsAlt>;<\n>
<else>
 <ruleBacktrackFailure()>
 NoViableAltException
 nvae(input.getPosition(), "<description>", <decisionNumber>, <stateNumber>);<\n>
 <@noViableAltException()>
 throw nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<if(!semPredState)>
<tokenid_type()> LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<endif>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
 */
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<if(!semPredState)>
<tokenid_type()> LA<decisionNumber>_<stateNumber> = input.LA(<k>);
<endif>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
else
{
 alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
```

```
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>)
{
 <targetState>
}
>>

// Fixed DFA (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
 alt<decisionNumber> = <eotPredictsAlt>;
<else>
 NoViableAltException nvae( input.getPosition(), "<description>", <decisionNumber>, <stateNumber> );<\n>
 <@noViableAltException()>
 throw nvae;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
 <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
 alt<decisionNumber> = <eotPredictsAlt>;
 break;<\n>
<endif>
}<\n>
```

```
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n"> {
 <targetState>
} break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The
 <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
// dfaDecision
alt<decisionNumber> = predictDFA<decisionNumber>(input);
>>

/** The overall cyclic DFA chunk; contains all the DFA states */
cyclicDFA(dfa) ::= <<
/* cyclicDFA=<dfa>
*/
// cyclic    = <dfa.cyclic>
// numstates = <dfa.numberOfStates>

// startState = <dfa.startState>
// startState.numberOfTransitions = <dfa.startState.NumberOfTransitions>
// startState.lookaheadDepth = <dfa.startState.LookaheadDepth>

const static short <name>dfa<dfa.decisionNumber>_eot[<dfa.numberOfStates>] = {
 <dfa.eot; wrap="\n     ", separator=",", null="-1">
};
const static short <name>dfa<dfa.decisionNumber>_eof[<dfa.numberOfStates>] = {
 <dfa.eof; wrap="\n     ", separator=",", null="-1">
};
const static unichar <name>dfa<dfa.decisionNumber>_min[<dfa.numberOfStates>] = {
 <dfa.min; wrap="\n     ", separator=",", null="0">
};
const static unichar <name>dfa<dfa.decisionNumber>_max[<dfa.numberOfStates>]
 = {
 <dfa.max; wrap="\n     ", separator=",", null="0">
};
const static short <name>dfa<dfa.decisionNumber>_accept[<dfa.numberOfStates>] = {
 <dfa.accept; wrap="\n     ", separator=",", null="-1">
};
const static short <name>dfa<dfa.decisionNumber>_special[<dfa.numberOfStates>] = {
 <dfa.special; wrap="\n     ", separator=",", null="-1">
};
<dfa.edgeTransitionClassMap.keys:{ table |
const static short <name>dfa<dfa.decisionNumber>_transition<i0>[] = {
 <table; separator=", ", wrap="\n ", null="-1">
};
}; null="">
const static short <name>dfa<dfa.decisionNumber>_transition[] = {
 <dfa.transitionEdgeTables:{whichTable|<name>dfa<dfa.decisionNumber>_transition<whichTable>,};
separator="\n", null="0 /* fixme? */">
};
 <! add attribute for the DFA !>
 DFA\<char_type> dfa<dfa.decisionNumber>;
<! this should go in the initializer of the thing
- (id) init
{
 if ((self = [super init]) != nil) {
  eot = <name>dfa<dfa.decisionNumber>_eot;
  eof = <name>dfa<dfa.decisionNumber>_eof;
  min
 = <name>dfa<dfa.decisionNumber>_min;
  max = <name>dfa<dfa.decisionNumber>_max;
  accept = <name>dfa<dfa.decisionNumber>_accept;
  special = <name>dfa<dfa.decisionNumber>_special;
  if (!(transition = calloc(<dfa.numberOfStates>, sizeof(void*)))) {
   [self release];
   return nil;
  }
  <dfa.transitionEdgeTables:{whichTable|transition[<i0>] =
<name>dfa<dfa.decisionNumber>_transition<whichTable>;}; separator="\n", null="">
 }
 return self;
}
!>

<if(dfa.specialStateSTs)>
int specialStateTransition( int state )
{
 int s = state;
 switch ( s ) {
  <dfa.specialStateSTs:{state |
  case <i0> : <! compressed special state numbers 0..n-1 !>
  <state>}; separator="\n">
 }
<if(backtracking)>
 if ( recognizer.isBacktracking() ) {
  recognizer.setFailed();
  return -1;
 }<\n>
<endif>
 noViableAlt(s, input);
}<\n>
<endif>


<\n>

// <dfa.description>
decision_type predictDFA<dfa.decisionNumber>( StreamType& input )
{
 /* mark current location (rewind automatically when the rewinder goes
  * out
 of scope */
 antlr3::Rewinder\<position_type> markPoint(input.getPosition());
 goto s0; // goto start...
 // ...
 throw NoViableAltException( input.getPosition(), "<dfa.description>", <dfa.decisionNumber>, 0 /* fixme */ );<\n>
}<\n>
>>

/** A state in a cyclic DFA */
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
// cyclicDFAState
s<stateNumber>: {
 <if(semPredState)>
 input.rewind();<\n>
 <else>
 <tokenid_type()> LA<decisionNumber>_<stateNumber> = input.LA(1);
 <endif>
 <edges>
 <if(needErrorClause)>
 throw NoViableAltException( input.getPosition(), "<description>", <decisionNumber>, <stateNumber> );<\n>
 <endif><\n>
}<\n>
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
// cyclicDFAEdge
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>)
{
 input.consume();
 goto s<targetStateNumber>;
}<\n>
```

```
>>

/**
 An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= "goto s<targetStateNumber>;"

// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left> && <right>)"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o | ||<o>}>)"

notPredicate(pred) ::= "!(<pred>)"

evalPredicate(pred,description) ::= "<pred>"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber>\>=<lower> && LA<decisionNumber>_<stateNumber>\<=<upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt)
 ::= "(input.LA(<k>)\>=<lower> && input.LA(<k>)\<=<upper>)"

setTest(ranges) ::= "<ranges; separator=\"||\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name> {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
```

```
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.name>_return
<else>
<if(ruleDescriptor.singleValueReturnType)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.name>_return
<else>
<if(referencedRule.singleValueReturnType)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/**
 Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<javaTypeInitMap.(typeName)>
>>

ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public static class <returnType()> {
   <labelType> start, stop;
<if(buildAST)>
   <ASTLabelType> tree;
<else>
<if(buildTemplate)>
   StringTemplate st;
<endif>
<endif>
   <scope.attributes:{<it.decl>;}; separator="\n">
};
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

/** Used in codegen.g to translate $x.y references.
*  I could have left actions as StringTemplates to be inserted in
*  the output (so they could use attributes inherited from surrounding
*  templates), but
 really wanted to pass in AttributeScope and Attribute
*  objects so this translation could query them.  So, translation of
*  $x.y to executable code occurs before recognizerST.toString() occurs.
*  I.e., actions are just text strings during final code generation.
*/
globalAttributeRef(scope,attr) ::= <<
((<scope>)<scope>_stack.peek()).<attr.name>
>>

parameterAttributeRef(attr) ::= "<attr.name>"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name>
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
```

```
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute
 of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.singleValueReturnType)>
<scope>
<else>
<scope>.<attr.name>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.singleValueReturnType)>
<attr.name>
<else>
retval.<attr.name>
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"

// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>.getText()"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
tokenLabelPropertyRef_index(scope,attr) ::=
 "<scope>.getTokenIndex()"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr) ::= "input.toString(<scope>.start,<scope>.stop)"
ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "<scope>.getType()"
```

```
lexerRuleLabelPropertyRef_line(scope,attr) ::= "<scope>.getLine()"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "<scope>.getCharPositionInLine()"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "<scope>.getChannel()"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "<scope>.getTokenIndex()"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "<scope>.getText()"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr)
 ::= "((<labelType>)retval.start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType>)retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.tree)"
rulePropertyRef_text(scope,attr) ::= "input.toString(retval.start,input.LT(-1))"
rulePropertyRef_st(scope,attr) ::= "retval.st"

// A C T I O N S

emit(type) ::= "emit(<type>);"

setType(type) ::= "setType(<type>);"

/** How to execute an action */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> )
{
 <action>
}
<else>
if ( backtracking == 0 )
{
 <action>
}
<endif>
<else>
<action>
<endif>
>>

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static final BitSet <name> = new BitSet(new long[]{<words64:{<it>L};separator=",">});<\n>
>>

tokenPrefix() ::= "TOK_"
codeFileExtension() ::= ".cpp"
// used in CPPTarget.java to generate the headerfile extension
```

headerFileExtension() ::= ".h"

true() ::= "true"
false()
 ::= "false"

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CPP/CPP.stg
No license file was found, but licenses were detected in source scan.

/** ANTLR's code generator.
*
* Generate recognizers derived from grammars. Language independence
* achieved through the use of StringTemplateGroup objects. All output
* strings are completely encapsulated in the group files such as Java.stg.
* Some computations are done that are unused by a particular language.
* This generator just computes and sets the values into the templates;
* the templates are free to use or not use the information.
*

* To make a new code generation target, define X.stg for
 language X
* by copying from existing Y.stg most closely releated to your language;
* e.g., to do CSharp.stg copy Java.stg. The template group file has a
* bunch of templates that are needed by the code generator. You can add
* a new target w/o even recompiling ANTLR itself. The language=X option
* in a grammar file dictates which templates get loaded/used.
*
* Some language like C need both parser files and header files. Java needs
* to have a separate file for the cyclic DFA as ANTLR generates bytecodes
* directly (which cannot be in the generated parser Java file). To facilitate
* this,
*
* cyclic can be in same file, but header, output must be searpate. recognizer
* is in outptufile.
*/

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/CodeGenerator.java
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

```
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Templates for building ASTs during normal parsing.
 *
 *  Deal with many combinations.  Dimensions are:
 *  Auto build or rewrite
 *    no label, label, list label  (label/no-label handled together)
 *    child, root
 *    token, set, rule, wildcard
 *
 *  The situation is not too bad as rewrite (->) usage makes ^ and !
 *  invalid. There is no huge explosion of combinations.
 */
group ASTParser;

finishedBacktracking(block) ::= <<
<if(backtracking)>
if <actions.(actionScope).synpredgate>:
    <block>
<else>
<block>
<endif>
>>

@ruleBody.setErrorReturnValue() ::= <<
retval.tree = self._adaptor.errorNode(self.input, retval.start, self.input.LT(-1), re)
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<finishedBacktracking({
<label>_tree = <createNodeFromToken(...)>
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,hetero,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,hetero,elementIndex,postmatchCode) ::=
 <<
<super.matchSet(..., postmatchCode={<finishedBacktracking({self._adaptor.addChild(root_0,
<createNodeFromToken(...)>)})>})>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"
```

```
// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
<super.matchSet(..., postmatchCode={<finishedBacktracking({root_0 =
self._adaptor.becomeRoot(<createNodeFromToken(...)>, root_0)})>})>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<finishedBacktracking({self._adaptor.addChild(root_0, <label>.tree)})>
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope)
 ::= <<
<super.ruleRef(...)>
<finishedBacktracking({root_0 = self._adaptor.becomeRoot(<label>.tree, root_0)})>
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".tree",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem=label+".tree",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".tree",...)>
>>

// WILDCARD AST
```

```
wildcard(label,elementIndex) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree = self._adaptor.createWithPayload(<label>)
self._adaptor.addChild(root_0, <label>_tree)
})>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<finishedBacktracking({
<label>_tree
 = self._adaptor.createWithPayload(<label>)
root_0 = self._adaptor.becomeRoot(<label>_tree, root_0)
})>
>>

createNodeFromToken(label,hetero) ::= <<
<if(hetero)>
<hetero>(<label>) <! new MethodNode(IDLabel) !>
<else>
self._adaptor.createWithPayload(<label>)
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<finishedBacktracking({
retval.tree = self._adaptor.rulePostProcessing(root_0)
self._adaptor.setTokenBoundaries(retval.tree, retval.start, retval.stop)
})>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/ASTParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2006, 2007 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions

are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
#import \<ANTLR/ANTLR.h><\n>
<endif>
>>

@parserHeaderFile.ivars() ::= <<
<@super.ivars()>
<parserIVars()>
>>

@parserHeaderFile.methodsdecl() ::= <<
<@super.methodsdecl()>
<parserMethodsDecl()>
>>

@genericParser.methods() ::= <<
<@super.methods()>
<parserMethods()>
>>

/** additional init code for tree support */
@genericParser.init() ::= <<
<@super.init()>
```

```
[self
 setTreeAdaptor:[[[ANTLRCommonTreeAdaptor alloc] init] autorelease]];
>>



@genericParser.dealloc() ::= <<
[self setTreeAdaptor:nil];
<@super.dealloc()>
>>

/** Add an adaptor property that knows how to build trees */
parserIVars() ::= <<
id\<ANTLRTreeAdaptor> treeAdaptor;
>>

/** Declaration of additional tree support methods - go in interface of parserHeaderFile() */
parserMethodsDecl() ::= <<
- (id\<ANTLRTreeAdaptor>) treeAdaptor;
- (void) setTreeAdaptor:(id\<ANTLRTreeAdaptor>)theTreeAdaptor;
>>

/** Definition of addition tree support methods - go in implementation of genericParser() */
parserMethods() ::= <<
- (id\<ANTLRTreeAdaptor>) treeAdaptor
{
 return treeAdaptor;
}

- (void) setTreeAdaptor:(id\<ANTLRTreeAdaptor>)aTreeAdaptor
{
 if (aTreeAdaptor != treeAdaptor) {
  [aTreeAdaptor retain];
  [treeAdaptor release];
  treeAdaptor = aTreeAdaptor;
 }
}
>>

/** addition ivars for returnscopes */
@returnScopeInterface.ivars() ::= <<
<recognizer.ASTLabelType; null="id"> tree;
>>

/**
 the interface of returnScope methods */
@returnScopeInterface.methods() ::= <<
- (<recognizer.ASTLabelType; null="id">) tree;
- (void) setTree:(<recognizer.ASTLabelType; null="id">)aTree;
```

```
>>

/** the implementation of returnScope methods */
@returnScope.methods() ::= <<
- (<ASTLabelType>) tree
{
   return tree;
}
- (void) setTree:(<ASTLabelType>)aTree
{
   if (tree != aTree) {
       [aTree retain];
       [tree release];
       tree = aTree;
   }
}

- (void) dealloc
{
   [self setTree:nil];
   [super dealloc];
}
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = nil;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{<ASTLabelType> _<it.label.text>_tree = nil;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<ASTLabelType> _<it.label.text>_tree = nil;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{ANTLRRewriteRuleTokenStream *_stream_<it>=[[ANTLRRewriteRuleTokenStream
 alloc] initWithTreeAdaptor:treeAdaptor description:@"token <it>"];}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{ANTLRRewriteRuleSubtreeStream *_stream_<it>=[[ANTLRRewriteRuleSubtreeStream alloc]
initWithTreeAdaptor:treeAdaptor description:@"rule <it>"];}; separator="\n">

>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<[ruleDescriptor.allTokenRefsInAltsWithRewrites,ruleDescriptor.allRuleRefsInAltsWithRewrites]
   :{[_stream_<it> release];}; separator="\n">
<if(ruleDescriptor.hasMultipleReturnValues)>
```

```
<if(backtracking)>
if ( ![_state isBacktracking] ) {<\n>
<endif>
   [_<prevRuleRootRef()> setTree:(<ASTLabelType>)[treeAdaptor postProcessTree:root_0]];
   [treeAdaptor setBoundariesForTree:[_<prevRuleRootRef()> tree] fromToken:[_<prevRuleRootRef()> start]
toToken:[_<prevRuleRootRef()> stop]];
<if(backtracking)>
}
<endif>
<endif>
[root_0 release];
>>

rewriteCodeLabelsCleanUp() ::= <<
<referencedTokenLabels
   :{[_stream_<it> release];};
    separator="\n"
>
<referencedTokenListLabels
   :{[_stream_<it> release];};
   separator="\n"
>
<referencedRuleLabels
   :{[_stream_<it> release];};
   separator="\n"
>
<referencedRuleListLabels
   :{[_stream_<it> release];};
   separator="\n"
>
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
root_0 = (<ASTLabelType>)[treeAdaptor newEmptyTree];<\n>
<endif>
<endif>
>>

// T r a c k i n g  R u l e  E l e m e n t s

/** ID and track it for use in a rewrite rule */
```

```
tokenRefTrack(token,label,elementIndex) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( ![_state isBacktracking] ) <endif>[_stream_<token> addElement:_<label>];<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID
 for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( ![_state isBacktracking] ) <endif>[_stream_<token> addElement:_<label>];<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( ![_state isBacktracking] ) <endif>[_stream_<rule.name> addElement:[_<label> tree]];
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope)
 ::= <<
<super.ruleRefRuleRoot(...)>
<if(backtracking)>if ( ![_state isBacktracking] ) <endif>[_stream_<rule.name> addElement:[_<label> tree]];<\n>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
```

```
<listLabel(elem="["+label+" tree]",...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if (![_state isBacktracking])
 {<\n>
<endif>
int i_0 = 0;
root_0 = (<ASTLabelType>)[treeAdaptor newEmptyTree];
[_<prevRuleRootRef()> setTree:root_0];
<rewriteCodeLabels()>
<alts:rewriteAlt(); separator="else ">
<rewriteCodeLabelsCleanUp()>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{ANTLRRewriteRuleTokenStream *_stream_<it>=[[ANTLRRewriteRuleTokenStream alloc]
initWithTreeAdaptor:treeAdaptor description:@"token <it>" element:_<it>];};
   separator="\n"
>
<referencedTokenListLabels
   :{ANTLRRewriteRuleTokenStream *_stream_<it>=[[ANTLRRewriteRuleTokenStream alloc]
initWithTreeAdaptor:treeAdaptor description:@"token <it>" elements:_<it>_list];};
   separator="\n"
>
<referencedRuleLabels
```

```
   :{ANTLRRewriteRuleSubtreeStream *_stream_<it>=[[ANTLRRewriteRuleSubtreeStream alloc]
initWithTreeAdaptor:treeAdaptor description:@"token <it>" element:_<it>!=nil?[_<it> tree]:nil];};
   separator="\n"
>
<referencedRuleListLabels
   :{ANTLRRewriteRuleSubtreeStream *_stream_<it>=[[ANTLRRewriteRuleSubtreeStream
 alloc] initWithTreeAdaptor:treeAdaptor description:@"token <it>" elements:_list_<it>];};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | [_stream_<el> hasNext]}; separator="||"> ) {
 <alt>
}
<referencedElementsDeep:{el | [_stream_<el> reset];<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | [_stream_<el> hasNext]}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | [_stream_<el> reset];<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep,
 // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
```

```
{
if ( !(<referencedElements:{el | [_stream_<el> hasNext]}; separator="||">) ) {
   @throw [NSException exceptionWithName:@"RewriteEarlyExitException" reason:nil userInfo:nil];
}
while ( <referencedElements:{el | [_stream_<el> hasNext]}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | [_stream_<el> reset];<\n>}>
}
>>



rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = nil;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)[treeAdaptor newEmptyTree];
<root:rewriteElement()>
<children:rewriteElement()>
[treeAdaptor
 addChild:root_<treeLevel> toTree:root_<enclosingTreeLevel>];
[root_<treeLevel> release];
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
```

```
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
<if(args)>
id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithToken:_<token>]; // TODO: args: <args;
separator=", ">
<endif>
[treeAdaptor addChild:<if(args)>_<token>_tree<else>[_stream_<token> next]<endif> toTree:root_<treeLevel>];
<if(args)>
[_<token>_tree release];<\n>
<endif>
<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
[treeAdaptor addChild:[_stream_<label> next] toTree:root_<treeLevel>];<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
[treeAdaptor addChild:[_stream_<label> next] toTree:root_<treeLevel>];<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex)
 ::= <<
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:[_stream_<label> next]
parentOf:root_<treeLevel>];<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:[_stream_<token> next]
parentOf:root_<treeLevel>];<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
<if(first(rest(args)))><! got two arguments - means create from token with custom text!>
id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithToken:<first(args)> tokenType:<token>
text:@<first(rest(args))>];
[treeAdaptor addChild:_<token>_tree toTree:root_<treeLevel>];
[_<token>_tree release];<\n>
<else><! at most one argument !>
<if(first(args))>
id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithToken:<first(args)> tokenType:<token>];
[treeAdaptor addChild:_<token>_tree
```

```
 toTree:root_<treeLevel>];
[_<token>_tree release];<\n>
<else><! no argument at all !>
id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithTokenType:<token> text:[tokenNames
objectAtIndex:<token>]];
[treeAdaptor addChild:_<token>_tree toTree:root_<treeLevel>];
[_<token>_tree release];<\n>
<endif><! one arg !>
<endif><! two args !>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
<if(first(rest(args)))><! got two arguments - means create from token with custom text!>
id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithToken:<first(args)> tokenType:<token>
text:@<first(rest(args))>];
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:_<token>_tree parentOf:root_<treeLevel>];
[_<token>_tree release];<\n>
<else><! at most one argument !>
<if(first(args))>
id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithToken:<first(args)> tokenType:<token>];
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:_<token>_tree parentOf:root_<treeLevel>];
[_<token>_tree
 release];<\n>
<else><! no argument at all !>id\<ANTLRTree> _<token>_tree = [treeAdaptor newTreeWithTokenType:<token>
text:[tokenNames objectAtIndex:<token>]];
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:_<token>_tree parentOf:root_<treeLevel>];
[_<token>_tree release];<\n>
<endif><! one arg !>
<endif><! two args !>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
[treeAdaptor addChild:[_stream_<rule> next] toTree:root_<treeLevel>];<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
```

```
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:(id\<ANTLRTree>)[_stream_<rule> next]
parentOf:root_<treeLevel>];<\n>
>>

rewriteNodeAction(action)
 ::= <<
[treeAdaptor addChild:<action> toTree:root_<treeLevel>];<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:<action> parentOf:root_<treeLevel>];<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
[treeAdaptor addChild:[_<label> tree] toTree:root_<treeLevel>];<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
[treeAdaptor addChild:[(ANTLR<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope
*)[_stream_<label> next] tree] toTree:root_<treeLevel>];<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor makeNode:[_<label> tree] parentOf:root_<treeLevel>];<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)[treeAdaptor
 makeNode:[(ANTLR<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope *)[_stream_<label> next]
tree] parentOf:root_<treeLevel>];<\n>
>>

createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel, args) !>
new <hetero>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.nextToken()<if(args)>, <args; separator=", "><endif>)
```

```
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ObjC/AST.stg

No license file was found, but licenses were detected in source scan.

```
/*
New style messages. This file contains the actual layout of the messages emitted by ANTLR.
The text itself is coming out of the languages/*stg files, according to the chosen locale.
This file contains the default format ANTLR uses.
*/

group antlr;
```

location(file, line, column) ::= "<file>:<line>:<column>:"

message(id, text) ::= "(<id>) <text>"

report(location, message, type) ::= "<type>(<message.id>): <location> <message.text>"

wantsSingleLineMessage() ::= "false"

Found in path(s):
*
 /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/templates/messages/formats/antlr.stg
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD licence"]
* Copyright (c) 2005-2008 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2008-2009 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
 OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*
  invalid. There is no huge explosion of combinations.
*/
group ASTParser;

@rule.setErrorReturnValue() ::= <<
retval.tree = (<ASTLabelType>)adaptor.ErrorNode(input, retval.start, input.LT(-1), re);
<! System.out.WriteLine("<ruleName> returns "+((CommonTree)retval.tree).toStringTree()); !>
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = <createNodeFromToken(...)>;
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
>>

/**
 ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>
```

```
/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,hetero,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root
 code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
<super.matchSet(..., postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> )
<endif>adaptor.AddChild(root_0, <createNodeFromToken(...)>);})>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<super.matchSet(..., postmatchCode={<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0
= (<ASTLabelType>)adaptor.BecomeRoot(<createNodeFromToken(...)>, root_0);})>
>>
```

```
//
 RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>adaptor.AddChild(root_0, <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_0);
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

// WILDCARD AST

wildcard(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>}<endif>
>>
```

```
wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>}<endif>
>>

createNodeFromToken(label,hetero) ::= <<
<if(hetero)>
new <hetero>(<label>) <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType>)adaptor.Create(<label>)
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree =
 (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
adaptor.SetTokenBoundaries(retval.tree, retval.start, retval.stop);
<if(backtracking)>}<endif>
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/ASTParser.stg

No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD licence"]
* Copyright (c) 2005-2008 Terence Parr
* All rights reserved.
*
* Conversion to C#:
* Copyright (c) 2008-2009 Sam Harwell, Pixel Mine, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
 OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/
group ASTDbg;

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor;
public ITreeAdaptor TreeAdaptor
{
 get
 {
  return adaptor;
 }
 set
 {
<if(grammar.grammarIsRoot)>
  this.adaptor
 = new DebugTreeAdaptor(dbg,adaptor);
<else>
  this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
  <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
}<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>
```

```
createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy = new DebugEventSocketProxy( this, port,
<if(TREE_PARSER)>input.TreeAdaptor<else>adaptor<endif> );
DebugListener = proxy;
<inputStreamType> = new Debug<inputStreamType>( input, proxy );
try
{
 proxy.Handshake();
}
catch ( IOException ioe )
{
 ReportError( ioe );
}
>>

@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

@ctorForProfilingRootGrammar.finally() ::=<<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

@ctorForPredefinedListener.superClassRef() ::= ": base( input, dbg )"

@ctorForPredefinedListener.finally()
 ::=<<
<if(grammar.grammarIsRoot)><! don't create new adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>

@rewriteElement.pregen() ::= "dbg.Location( <e.line>, <e.pos> );"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com

```
/*
* This code generating template and the associated C runtime was produced by:
* Jim Idle jimi|hereisanat|idle|dotgoeshere|ws.
* If it causes the destruction of the Universe, it will be pretty cool so long as
* I am in a different one at the time.
*/
group C implements ANTLRCore ;

cTypeInitMap ::= [
 "int"      : "0",             // Integers     start out being 0
 "long"      : "0",
         // Longs        start out being 0
 "float"      : "0.0",           // Floats       start out being 0
 "double"     : "0.0",           // Doubles      start out being 0
 "ANTLR3_BOOLEAN"    : "ANTLR3_FALSE",   // Booleans     start out being Antlr C for false
 "byte"      : "0",             // Bytes        start out being 0
 "short"      : "0",             // Shorts       start out being 0
 "char"      : "0",             // Chars        start out being 0
 default      : "NULL"           // Anything other than an atomic type (above) is a NULL (probably NULL pointer).
]
```

```
leadIn(type) ::=
<<
/** \file
*  This <type> file was generated by $ANTLR version <ANTLRVersion>
*
*     -  From the grammar source file : <fileName>
*     -                            On : <generatedTimestamp>
<if(LEXER)>
*     -                 for the lexer : <name>Lexer
<endif>
<if(PARSER)>
*     -                for the parser : <name>Parser
<endif>
<if(TREE_PARSER)>
*     -           for the tree
 parser : <name>TreeParser
<endif>
*
* Editing it, at least manually, is not wise.
*
* C language generator and runtime by Jim Idle, jimi|hereisanat|idle|dotgoeshere|ws.
*
*
>>

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile( LEXER,
            PARSER,
            TREE_PARSER,
            actionScope,
            actions,
            docComment,
            recognizer,
            name,
            tokens,
            tokenNames,
            rules,
            cyclicDFAs,
            bitsets,
            buildTemplate,
            buildAST,
            rewriteMode,
            profile,
            backtracking,
            synpreds,
            memoize,
            numRules,
            fileName,
            ANTLRVersion,
            generatedTimestamp,
            trace,
            scopes,
            superClass,
            literals
            ) ::=
<<
<leadIn("C source")>
*/
// [The "BSD licence"]
//
 Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND
 FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<if(actions.(actionScope).header)>

/* =============================================================================
* This is what the grammar programmer asked us to put at the top of every file.
```

```
*/
<actions.(actionScope).header>
/* End of Header action.
* =============================================================================
*/
<endif>

/* -----------------------------------------
* Include the ANTLR3 generated header file.
 */
#include    "<name>.h"
<actions.(actionScope).postinclude>
/* ----------------------------------------- */

<docComment>

<if(literals)>
/** String literals used by <name> that we must do things like MATCHS() with.
*  C will normally just lay down 8 bit characters, and you can use L"xxx" to
*  get wchar_t, but wchar_t is 16 bits on Windows, which is not UTF32 and so
*  we perform this little trick of defining the literals as arrays of UINT32
*  and passing in the address of these.
*/
<literals:{static ANTLR3_UCHAR lit_<i>[]  = <it>;}; separator="\n">

<endif>




/* MACROS that hide the C interface implementations from the
* generated code, which makes it a little more understandable to the human eye.
* I am very much against using C pre-processor macros for function calls and bits
* of code as you cannot see what is happening when single stepping in debuggers
* and so on. The exception (in my book at least) is for generated code, where you are
* not maintaining it, but
 may wish to read and understand it. If you single step it, you know that input()
* hides some indirect calls, but is always referring to the input stream. This is
* probably more readable than ctx->input->istream->input(snarfle0->blarg) and allows me to rejig
* the runtime interfaces without changing the generated code too often, without
* confusing the reader of the generated output, who may not wish to know the gory
* details of the interface inheritance.
*/

#define  CTX ctx

/* Aids in accessing scopes for grammar programmers
*/
```

```
#undef SCOPE_TYPE
#undef SCOPE_STACK
#undef SCOPE_TOP
#define SCOPE_TYPE(scope)   p<name>_##scope##_SCOPE
#define SCOPE_STACK(scope)  p<name>_##scope##Stack
#define SCOPE_TOP(scope)    ctx->p<name>_##scope##Top
#define SCOPE_SIZE(scope)   (ctx->SCOPE_STACK(scope)->size(ctx->SCOPE_STACK(scope)))
#define SCOPE_INSTANCE(scope, i) (ctx->SCOPE_STACK(scope)->get(ctx->SCOPE_STACK(scope),i))

<if(LEXER)>

/* Macros for accessing things in a lexer
*/
#undef
   LEXER
#undef    RECOGNIZER
#undef    RULEMEMO
#undef    GETCHARINDEX
#undef    GETLINE
#undef    GETCHARPOSITIONINLINE
#undef    EMIT
#undef    EMITNEW
#undef    MATCHC
#undef    MATCHS
#undef    MATCHRANGE
#undef    LTOKEN
#undef    HASFAILED
#undef    FAILEDFLAG
#undef    INPUT
#undef    STRSTREAM
#undef    LA
#undef    HASEXCEPTION
#undef    EXCEPTION
#undef    CONSTRUCTEX
#undef    CONSUME
#undef    LRECOVER
#undef    MARK
#undef    REWIND
#undef    REWINDLAST
#undef    BACKTRACKING
#undef  MATCHANY
#undef  MEMOIZE
#undef  HAVEPARSEDRULE
#undef  GETTEXT
#undef  INDEX
#undef  SEEK
#undef  PUSHSTREAM
#undef  POPSTREAM
```

```
#undef  SETTEXT
#undef  SETTEXT8

#define     LEXER     ctx->pLexer
#define     RECOGNIZER       LEXER->rec
#define  LEXSTATE    RECOGNIZER->state
#define  TOKSOURCE    LEXSTATE->tokSource
#define     GETCHARINDEX()   LEXER->getCharIndex(LEXER)
#define     GETLINE()    LEXER->getLine(LEXER)
#define     GETTEXT()    LEXER->getText(LEXER)
#define
   GETCHARPOSITIONINLINE() LEXER->getCharPositionInLine(LEXER)
#define     EMIT()     LEXSTATE->type = _type; LEXER->emit(LEXER)
#define     EMITNEW(t)    LEXER->emitNew(LEXER, t)
#define     MATCHC(c)    LEXER->matchc(LEXER, c)
#define     MATCHS(s)    LEXER->matchs(LEXER, s)
#define     MATCHRANGE(c1,c2)     LEXER->matchRange(LEXER, c1, c2)
#define     MATCHANY()    LEXER->matchAny(LEXER)
#define     LTOKEN      LEXSTATE->token
#define     HASFAILED()    (LEXSTATE->failed == ANTLR3_TRUE)
#define     BACKTRACKING   LEXSTATE->backtracking
#define     FAILEDFLAG    LEXSTATE->failed
#define     INPUT     LEXER->input
#define     STRSTREAM    INPUT
#define  ISTREAM     INPUT->istream
#define  INDEX()     ISTREAM->index(ISTREAM)
#define  SEEK(n)     ISTREAM->seek(ISTREAM, n)
#define     EOF_TOKEN    &(LEXSTATE->tokSource->eofToken)
#define     HASEXCEPTION()   (LEXSTATE->error == ANTLR3_TRUE)
#define     EXCEPTION    LEXSTATE->exception
#define     CONSTRUCTEX()   RECOGNIZER->exConstruct(RECOGNIZER)
#define
   LRECOVER()    LEXER->recover(LEXER)
#define     MARK()     ISTREAM->mark(ISTREAM)
#define     REWIND(m)    ISTREAM->rewind(ISTREAM, m)
#define     REWINDLAST()   ISTREAM->rewindLast(ISTREAM)
#define  MEMOIZE(ri,si)   RECOGNIZER->memoize(RECOGNIZER, ri, si)
#define  HAVEPARSEDRULE(r)  RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define  PUSHSTREAM(str)   LEXER->pushCharStream(LEXER, str)
#define  POPSTREAM()    LEXER->popCharStream(LEXER)
#define  SETTEXT(str)   LEXSTATE->text = str
#define  SKIP()     LEXSTATE->token = &(TOKSOURCE->skipToken)
#define  USER1     LEXSTATE->user1
#define  USER2     LEXSTATE->user2
#define  USER3     LEXSTATE->user3
#define  CUSTOM     LEXSTATE->custom
#define  RULEMEMO    LEXSTATE->ruleMemo
#define  DBG      RECOGNIZER->debugger
```

```
/* If we have been told we can rely on the standard 8 bit or 16 bit input
* stream, then we can define our macros to use the direct pointers
* in the input object, which is much faster
 than indirect calls. This
* is really only significant to lexers with a lot of fragment rules (which
* do not place LA(1) in a temporary at the moment) and even then
* only if there is a lot of input (order of say 1M or so).
*/
#if defined(ANTLR3_INLINE_INPUT_ASCII) || defined(ANTLR3_INLINE_INPUT_UTF16)

# ifdef ANTLR3_INLINE_INPUT_ASCII

/* 8 bit "ASCII" (actually any 8 bit character set) */

#  define     NEXTCHAR   ((pANTLR3_UINT8)(INPUT->nextChar))
#  define     DATAP    ((pANTLR3_UINT8)(INPUT->data))

# else

#  define     NEXTCHAR   ((pANTLR3_UINT16)(INPUT->nextChar))
#  define     DATAP    ((pANTLR3_UINT16)(INPUT->data))

# endif

# define     LA(n) ((NEXTCHAR + n) > (DATAP + INPUT->sizeBuf) ? ANTLR3_CHARSTREAM_EOF :
(ANTLR3_UCHAR)(*(NEXTCHAR + n - 1)))
# define     CONSUME()          \
{                \
   if (NEXTCHAR \< (DATAP + INPUT->sizeBuf))      \
   {               \
  INPUT->charPositionInLine++;        \
  if  ((ANTLR3_UCHAR)(*NEXTCHAR) == INPUT->newlineChar)  \
  {              \
   INPUT->line++;          \
   INPUT->charPositionInLine =
 0;      \
   INPUT->currentLine  = (void *)(NEXTCHAR + 1);  \
  }              \
  INPUT->nextChar = (void *)(NEXTCHAR + 1);     \
   }               \
}

#else

// Pick up the input character by calling the input stream implementation.
//
#define     CONSUME()    INPUT->istream->consume(INPUT->istream)
```

```
#define    LA(n)    INPUT->istream->_LA(INPUT->istream, n)

#endif
<endif>

<if(PARSER)>
/* Macros for accessing things in the parser
*/

#undef    PARSER
#undef    RECOGNIZER
#undef    HAVEPARSEDRULE
#undef  MEMOIZE
#undef    INPUT
#undef    STRSTREAM
#undef    HASEXCEPTION
#undef    EXCEPTION
#undef    MATCHT
#undef    MATCHANYT
#undef    FOLLOWSTACK
#undef    FOLLOWPUSH
#undef    FOLLOWPOP
#undef    PRECOVER
#undef    PREPORTERROR
#undef    LA
#undef    LT
#undef    CONSTRUCTEX
#undef    CONSUME
#undef    MARK
#undef    REWIND
#undef    REWINDLAST
#undef
   PERRORRECOVERY
#undef    HASFAILED
#undef    FAILEDFLAG
#undef    RECOVERFROMMISMATCHEDSET
#undef    RECOVERFROMMISMATCHEDELEMENT
#undef  INDEX
#undef     ADAPTOR
#undef  SEEK
#undef    RULEMEMO
#undef  DBG

#define    PARSER      ctx->pParser
#define    RECOGNIZER     PARSER->rec
#define  PSRSTATE     RECOGNIZER->state
#define    HAVEPARSEDRULE(r)    RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define  MEMOIZE(ri,si)     RECOGNIZER->memoize(RECOGNIZER, ri, si)
```

```
#define     INPUT       PARSER->tstream
#define     STRSTREAM      INPUT
#define  ISTREAM       INPUT->istream
#define  INDEX()       ISTREAM->index(INPUT->istream)
#define     HASEXCEPTION()     (PSRSTATE->error == ANTLR3_TRUE)
#define     EXCEPTION      PSRSTATE->exception
#define     MATCHT(t, fs)     RECOGNIZER->match(RECOGNIZER, t, fs)
#define     MATCHANYT()      RECOGNIZER->matchAny(RECOGNIZER)
#define     FOLLOWSTACK         PSRSTATE->following
#define     FOLLOWPUSH(x)     FOLLOWSTACK->push(FOLLOWSTACK,
 ((void *)(&(x))), NULL)
#define     FOLLOWPOP()      FOLLOWSTACK->pop(FOLLOWSTACK)
#define     PRECOVER()      RECOGNIZER->recover(RECOGNIZER)
#define     PREPORTERROR()     RECOGNIZER->reportError(RECOGNIZER)
#define     LA(n)       INPUT->istream->_LA(ISTREAM, n)
#define     LT(n)       INPUT->_LT(INPUT, n)
#define     CONSTRUCTEX()     RECOGNIZER->exConstruct(RECOGNIZER)
#define     CONSUME()      ISTREAM->consume(ISTREAM)
#define     MARK()       ISTREAM->mark(ISTREAM)
#define     REWIND(m)      ISTREAM->rewind(ISTREAM, m)
#define     REWINDLAST()     ISTREAM->rewindLast(ISTREAM)
#define  SEEK(n)       ISTREAM->seek(ISTREAM, n)
#define     PERRORRECOVERY     PSRSTATE->errorRecovery
#define     FAILEDFLAG      PSRSTATE->failed
#define     HASFAILED()      (FAILEDFLAG == ANTLR3_TRUE)
#define     BACKTRACKING     PSRSTATE->backtracking
#define     RECOVERFROMMISMATCHEDSET(s)  RECOGNIZER->recoverFromMismatchedSet(RECOGNIZER,
s)
#define     RECOVERFROMMISMATCHEDELEMENT(e) RECOGNIZER-
>recoverFromMismatchedElement(RECOGNIZER,
 s)
#define     ADAPTOR                      ctx->adaptor
#define  RULEMEMO      PSRSTATE->ruleMemo
#define  DBG        RECOGNIZER->debugger

<endif>

<if(TREE_PARSER)>
/* Macros for accessing things in the parser
*/

#undef     PARSER
#undef     RECOGNIZER
#undef     HAVEPARSEDRULE
#undef     INPUT
#undef     STRSTREAM
#undef     HASEXCEPTION
#undef     EXCEPTION
```

```
#undef    MATCHT
#undef    MATCHANYT
#undef    FOLLOWSTACK
#undef    FOLLOWPUSH
#undef    FOLLOWPOP
#undef    PRECOVER
#undef    PREPORTERROR
#undef    LA
#undef    LT
#undef    CONSTRUCTEX
#undef    CONSUME
#undef    MARK
#undef    REWIND
#undef    REWINDLAST
#undef    PERRORRECOVERY
#undef    HASFAILED
#undef    FAILEDFLAG
#undef    RECOVERFROMMISMATCHEDSET
#undef    RECOVERFROMMISMATCHEDELEMENT
#undef    BACKTRACKING
#undef     ADAPTOR
#undef    RULEMEMO
#undef  SEEK
#undef  INDEX
#undef  DBG

#define    PARSER       ctx->pTreeParser
#define
   RECOGNIZER      PARSER->rec
#define  PSRSTATE      RECOGNIZER->state
#define    HAVEPARSEDRULE(r)    RECOGNIZER->alreadyParsedRule(RECOGNIZER, r)
#define    INPUT       PARSER->ctnstream
#define  ISTREAM       INPUT->tnstream->istream
#define    STRSTREAM      INPUT->tnstream
#define    HASEXCEPTION()     (PSRSTATE->error == ANTLR3_TRUE)
#define    EXCEPTION      PSRSTATE->exception
#define    MATCHT(t, fs)     RECOGNIZER->match(RECOGNIZER, t, fs)
#define    MATCHANYT()      RECOGNIZER->matchAny(RECOGNIZER)
#define    FOLLOWSTACK         PSRSTATE->following
#define    FOLLOWPUSH(x)     FOLLOWSTACK->push(FOLLOWSTACK, ((void *)(&(x))), NULL)
#define    FOLLOWPOP()      FOLLOWSTACK->pop(FOLLOWSTACK)
#define    PRECOVER()      RECOGNIZER->recover(RECOGNIZER)
#define    PREPORTERROR()     RECOGNIZER->reportError(RECOGNIZER)
#define    LA(n)       ISTREAM->_LA(ISTREAM, n)
#define    LT(n)       INPUT->tnstream->_LT(INPUT->tnstream, n)
#define    CONSTRUCTEX()     RECOGNIZER->exConstruct(RECOGNIZER)
#define
```

```
   CONSUME()      ISTREAM->consume(ISTREAM)
#define     MARK()       ISTREAM->mark(ISTREAM)
#define     REWIND(m)      ISTREAM->rewind(ISTREAM, m)
#define     REWINDLAST()     ISTREAM->rewindLast(ISTREAM)
#define     PERRORRECOVERY     PSRSTATE->errorRecovery
#define     FAILEDFLAG      PSRSTATE->failed
#define     HASFAILED()      (FAILEDFLAG == ANTLR3_TRUE)
#define     BACKTRACKING     PSRSTATE->backtracking
#define     RECOVERFROMMISMATCHEDSET(s)  RECOGNIZER->recoverFromMismatchedSet(RECOGNIZER,
s)
#define     RECOVERFROMMISMATCHEDELEMENT(e) RECOGNIZER-
>recoverFromMismatchedElement(RECOGNIZER, s)
#define     ADAPTOR                        INPUT->adaptor
#define  RULEMEMO      PSRSTATE->ruleMemo
#define  SEEK(n)       ISTREAM->seek(ISTREAM, n)
#define  INDEX()       ISTREAM->index(ISTREAM)
#define  DBG        RECOGNIZER->debugger



<endif>

#define  TOKTEXT(tok, txt)    tok, (pANTLR3_UINT8)txt

/* The 4 tokens defined below may well clash with
 your own #defines or token types. If so
* then for the present you must use different names for your defines as these are hard coded
* in the code generator. It would be better not to use such names internally, and maybe
* we can change this in a forthcoming release. I deliberately do not #undef these
* here as this will at least give you a redefined error somewhere if they clash.
*/
#define     UP     ANTLR3_TOKEN_UP
#define     DOWN    ANTLR3_TOKEN_DOWN
#define     EOR     ANTLR3_TOKEN_EOR
#define     INVALID ANTLR3_TOKEN_INVALID



/* =============================================================================
* Functions to create and destroy scopes. First come the rule scopes, followed
* by the global declared scopes.
*/

<rules: {r |<if(r.ruleDescriptor.ruleScope)>
<ruleAttributeScopeFuncDecl(scope=r.ruleDescriptor.ruleScope)>
<ruleAttributeScopeFuncs(scope=r.ruleDescriptor.ruleScope)>
<endif>}>

<recognizer.scopes:{<if(it.isDynamicGlobalScope)>
<globalAttributeScopeFuncDecl(scope=it)>
```

```
<globalAttributeScopeFuncs(scope=it)>
<endif>}>

/*
 ============================================================================= */

/* =============================================================================
* Start of recognizer
*/

<recognizer>

/* End of code
* =============================================================================
*/

>>
headerFileExtension() ::= ".h"

headerFile( LEXER,
          PARSER,
          TREE_PARSER,
          actionScope,
          actions,
          docComment,
          recognizer,
          name,
          tokens,
          tokenNames,
          rules,
          cyclicDFAs,
          bitsets,
          buildTemplate,
          buildAST,
          rewriteMode,
          profile,
          backtracking,
          synpreds,
          memoize,
          numRules,
          fileName,
          ANTLRVersion,
          generatedTimestamp,
          trace,

  scopes,
   superClass,
          literals
```

```
    ) ::=
<<
<leadIn("C header")>
<if(PARSER)>
* The parser <mainName()>
<endif>
<if(LEXER)>
* The lexer <mainName()>
<endif>
<if(TREE_PARSER)>
* The tree parser <mainName()>
<endif>
has the callable functions (rules) shown below,
* which will invoke the code for the associated rule in the source grammar
* assuming that the input stream is pointing to a token/text stream that could begin
* this rule.
*
* For instance if you call the first (topmost) rule in a parser grammar, you will
* get the results of a full parse, but calling a rule half way through the grammar will
* allow you to pass part of a full token stream to the parser, such as for syntax checking
* in editors and so on.
*
* The parser entry points are called indirectly (by function pointer to function) via
* a parser context typedef p<name>, which is returned from a call to <name>New().
*
<if(LEXER)>
* As this is a generated lexer, it is unlikely you will
 call it 'manually'. However
* the methods are provided anyway.
*
<endif>
* The methods in p<name> are  as follows:
*
* <rules: {r | <if(!r.ruleDescriptor.isSynPred)> - <headerReturnType(ruleDescriptor=r.ruleDescriptor,...)>
p<name>-><r.ruleDescriptor.name>(p<name>)<endif>}; separator="\n * ">
*
* The return type for any particular rule is of course determined by the source
* grammar file.
*/
// [The "BSD licence"]
// Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
// http://www.temporal-wave.com
// http://www.linkedin.com/in/jimidle
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following
 disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. The name of the author may not be used to endorse or promote products
//    derived from this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
// IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
// OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
// INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
// NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
// THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#ifndef _<name>_H
#define
 _<name>_H
<actions.(actionScope).preincludes>
/* =============================================================================
* Standard antlr3 C runtime definitions
*/
#include    \<antlr3.h>

/* End of standard antlr 3 runtime definitions
* =============================================================================
*/
<actions.(actionScope).includes>

#ifdef __cplusplus
extern "C" {
#endif

// Forward declare the context typedef so that we can use it before it is
// properly defined. Delegators and delegates (from import statements) are
// interdependent and their context structures contain pointers to each other
// C only allows such things to be declared if you pre-declare the typedef.
//
typedef struct <name>_Ctx_struct <name>, * p<name>;

<if(recognizer.grammar.delegates)>
// Include delegate definition header files
```

```
//
<recognizer.grammar.delegates: {g|#include \<<g.recognizerName>.h>}; separator="\n">

<endif>



<actions.(actionScope).header>

#ifdef ANTLR3_WINDOWS
//
 Disable: Unreferenced parameter,       - Rules with parameters that are not used
//          constant conditional,       - ANTLR realizes that a prediction is always true (synpred usually)
//          initialized but unused variable     - tree rewrite variables declared but not needed
//          Unreferenced local variable      - lexer rule declares but does not always use _type
//          potentially unitialized variable used   - retval always returned from a rule
//   unreferenced local function has been removed - susually getTokenNames or freeScope, they can go without
warnigns
//
// These are only really displayed at warning level /W4 but that is the code ideal I am aiming at
// and the codegen must generate some of these warnings by necessity, apart from 4100, which is
// usually generated when a parser rule is given a parameter that it does not use. Mostly though
// this is a matter of orthogonality hence I disable that one.
//
#pragma warning( disable : 4100
 )
#pragma warning( disable : 4101 )
#pragma warning( disable : 4127 )
#pragma warning( disable : 4189 )
#pragma warning( disable : 4505 )
#pragma warning( disable : 4701 )
#endif
<if(backtracking)>

/* ========================
* BACKTRACKING IS ENABLED
* ========================
*/
<endif>

<rules:{r |<headerReturnScope(ruleDescriptor=r.ruleDescriptor,...)>}>

<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeDecl(scope=it)><endif>}>
<rules:{r |<ruleAttributeScopeDecl(scope=r.ruleDescriptor.ruleScope)>}>
<if(recognizer.grammar.delegators)>
// Include delegator definition header files
//
<recognizer.grammar.delegators: {g|#include \<<g.recognizerName>.h>}; separator="\n">
```

```
<endif>

/** Context tracking structure for <mainName()>
*/
struct <name>_Ctx_struct
{
   /** Built in ANTLR3 context tracker contains all the generic elements
    *  required for context tracking.
    */
<if(PARSER)>
   pANTLR3_PARSER   pParser;
<endif>
<if(LEXER)>
   pANTLR3_LEXER    pLexer;
<endif>
<if(TREE_PARSER)>
    pANTLR3_TREE_PARSER     pTreeParser;
<endif>

<if(recognizer.grammar.delegates)>
 <recognizer.grammar.delegates:
        {g|p<g.recognizerName> <g:delegateName()>;}; separator="\n">
<endif>
<if(recognizer.grammar.delegators)>
 <recognizer.grammar.delegators:
        {g|p<g.recognizerName> <g:delegateName()>;}; separator="\n">
<endif>
<scopes:{<if(it.isDynamicGlobalScope)>
   <globalAttributeScopeDef(scope=it)>
<endif>}; separator="\n\n">
<rules: {r |<if(r.ruleDescriptor.ruleScope)>
   <ruleAttributeScopeDef(scope=r.ruleDescriptor.ruleScope)>
<endif>}>

<if(LEXER)>
   <rules:{r | <if(!r.ruleDescriptor.isSynPred)><headerReturnType(ruleDescriptor=r.ruleDescriptor)>
(*m<r.ruleDescriptor.name>) (struct <name>_Ctx_struct * ctx<if(r.ruleDescriptor.parameterScope)>,
<endif><r.ruleDescriptor.parameterScope:parameterScope(scope=it)>);<endif>}; separator="\n";>
<endif>
<if(!LEXER)>
   <rules:{r | <headerReturnType(ruleDescriptor=r.ruleDescriptor)> (*<r.ruleDescriptor.name>) (struct
 <name>_Ctx_struct * ctx<if(r.ruleDescriptor.parameterScope)>,
<endif><r.ruleDescriptor.parameterScope:parameterScope(scope=it)>);}; separator="\n";>
<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
   // Delegated rules
<recognizer.grammar.delegatedRules:{ruleDescriptor|
   <headerReturnType(ruleDescriptor)> (*<ruleDescriptor.name>)(struct <name>_Ctx_struct *
ctx<if(ruleDescriptor.parameterScope)>, <endif><ruleDescriptor.parameterScope:parameterScope(scope=it)>);};
separator="\n";>
<endif>
  const char * (*getGrammarFileName)();
  void    (*free)   (struct <name>_Ctx_struct * ctx);
  <@members>
  <@end>
  <actions.(actionScope).context>
};

// Function protoypes for the constructor functions that external translation units
// such as delegators and delegates may wish to call.
//
ANTLR3_API p<name> <name>New         (<inputType()> instream<recognizer.grammar.delegators:{g|,
p<g.recognizerName>
 <g:delegateName()>}>);
ANTLR3_API p<name> <name>NewSSD      (<inputType()> instream,
pANTLR3_RECOGNIZER_SHARED_STATE state<recognizer.grammar.delegators:{g|, p<g.recognizerName>
<g:delegateName()>}>);
<if(!recognizer.grammar.grammarIsRoot)>
extern pANTLR3_UINT8   <recognizer.grammar.composite.rootGrammar.recognizerName>TokenNames[];
<endif>


/** Symbolic definitions of all the tokens that the <grammarType()> will work with.
* \{
*
* Antlr will define EOF, but we can't use that as it it is too common in
* in C header files and that would be confusing. There is no way to filter this out at the moment
* so we just undef it here for now. That isn't the value we get back from C recognizers
* anyway. We are looking for ANTLR3_TOKEN_EOF.
*/
#ifdef EOF
#undef EOF
#endif
#ifdef Tokens
#undef Tokens
#endif
<tokens:{#define <it.name>      <it.type>}; separator="\n">
#ifdef EOF
#undef EOF
#define EOF ANTLR3_TOKEN_EOF
#endif

#ifndef TOKENSOURCE
#define TOKENSOURCE(lxr) lxr->pLexer->rec->state->tokSource
#endif
```

```
/*
 End of token definitions for <name>
* =============================================================================
*/
/** \} */

#ifdef __cplusplus
}
#endif

#endif

/* END - Note:Keep extra line feed to satisfy UNIX systems */

>>

inputType() ::=<<
<if(LEXER)>
pANTLR3_INPUT_STREAM
<endif>
<if(PARSER)>
pANTLR3_COMMON_TOKEN_STREAM
<endif>
<if(TREE_PARSER)>
pANTLR3_COMMON_TREE_NODE_STREAM
<endif>
>>

grammarType() ::= <<
<if(PARSER)>
parser
<endif>
<if(LEXER)>
lexer
<endif>
<if(TREE_PARSER)>
tree parser
<endif>
>>

mainName() ::= <<
<if(PARSER)>
<name>
<endif>
<if(LEXER)>
<name>
<endif>
<if(TREE_PARSER)>
```

```
<name>
<endif>
>>

headerReturnScope(ruleDescriptor) ::= "<returnScope(...)>"

headerReturnType(ruleDescriptor) ::= <<
<if(LEXER)>
<if(!r.ruleDescriptor.isSynPred)>
void
<else>
<ruleDescriptor:returnType()>
<endif>
<else>
<ruleDescriptor:returnType()>
<endif>
>>

// Produce the lexer output
//
lexer(  grammar,
  name,
       tokens,
       scopes,
        rules,
       numRules,
       labelType="pANTLR3_COMMON_TOKEN",
       filterMode,
       superClass) ::= <<

<if(filterMode)>
/* Forward declare implementation function for ANTLR3_TOKEN_SOURCE interface when
* this is a filter mode lexer.
*/
static pANTLR3_COMMON_TOKEN <name>NextToken   (pANTLR3_TOKEN_SOURCE toksource);

/* Override the normal MEMOIZE and HAVEALREADYPARSED macros as this is a filtering
* lexer. In filter mode, the memoizing and backtracking are gated at BACKTRACKING > 1 rather
* than just BACKTRACKING. In some cases this might generate code akin to:
*   if (BACKTRACKING) if (BACKTRACKING > 1) memoize.
* However, I assume that the C compilers/optimizers are smart enough to work this one out
* these days - Jim
*/
#undef  MEMOIZE
#define  MEMOIZE(ri,si)   if (BACKTRACKING>1) { RECOGNIZER->memoize(RECOGNIZER, ri, si) }
#undef  HAVEPARSEDRULE
#define  HAVEPARSEDRULE(r)  if (BACKTRACKING>1) { RECOGNIZER->alreadyParsedRule(RECOGNIZER,
r) }
```

```
<endif>

/*
 Forward declare the locally static matching functions we have generated and any predicate functions.
*/
<rules:{r | static ANTLR3_INLINE <headerReturnType(ruleDescriptor=r.ruleDescriptor)>
<if(!r.ruleDescriptor.isSynPred)>m<endif><r.ruleDescriptor.name>    (p<name>
ctx<if(r.ruleDescriptor.parameterScope)>, <endif><r.ruleDescriptor.parameterScope:parameterScope(scope=it)>);};
separator="\n";>
static void <name>Free(p<name> ctx);

/* =========================================================================
* Lexer matching rules end.
* =========================================================================
*/

<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>

<actions.lexer.members>

static void
<name>Free  (p<name> ctx)
{
<if(memoize)>
 if (RULEMEMO != NULL)
 {
  RULEMEMO->free(RULEMEMO);
  RULEMEMO = NULL;
 }
<endif>
<if(grammar.directDelegates)>
 // Free the lexers that we delegated to
 // functions to. NULL the state so we only free it once.
 //
 <grammar.directDelegates:
        {g|ctx-><g:delegateName()>->pLexer->rec->state = NULL;
        ctx-><g:delegateName()>->free(ctx-><g:delegateName()>);}; separator="\n">
<endif>
   LEXER->free(LEXER);

   ANTLR3_FREE(ctx);
}

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
```

```
*/
static const char * getGrammarFileName()
{
 return fileName;
}

<if(filterMode)>
   <filteringNextToken()>
<endif>

/** \brief Create a new lexer called <name>
*
* \param[in]    instream Pointer to an initialized input stream
* \return
*     - Success p<name> initialized for the lex start
*     - Fail NULL
*/
ANTLR3_API p<name> <name>New
(<inputType()> instream<grammar.delegators:{g|, p<g.recognizerName> <g:delegateName()>}>)
{
 // See if we can create a new lexer with the standard constructor
 //
 return <name>NewSSD(instream, NULL<grammar.delegators:{g|,
 <g:delegateName()>}>);
}

/** \brief Create a new lexer called <name>
*
* \param[in]    instream Pointer to an initialized input stream
* \param[state] state Previously created shared recognizer stat
* \return
*     - Success p<name> initialized for the lex start
*     - Fail NULL
*/
ANTLR3_API p<name> <name>NewSSD
(pANTLR3_INPUT_STREAM instream, pANTLR3_RECOGNIZER_SHARED_STATE
state<grammar.delegators:{g|, p<g.recognizerName> <g:delegateName()>}>)
{
   p<name> ctx; // Context structure we will build and return

   ctx = (p<name>) ANTLR3_CALLOC(1, sizeof(<name>));

   if  (ctx == NULL)
   {
       // Failed to allocate memory for lexer context
       return  NULL;
   }
   /* -------------------------------------------------------------------
    * Memory for basic structure is allocated, now to fill in
    * in base ANTLR3 structures. We initialize the function pointers
    * for the standard ANTLR3 lexer function set, but upon return
    * from here,
 the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here so you can get a sense
    * of what goes where.
    */

   /* Create a base lexer, using the supplied input stream
    */
   ctx->pLexer = antlr3LexerNewStream(ANTLR3_SIZE_HINT, instream, state);

   /* Check that we allocated the memory correctly
    */
   if (ctx->pLexer == NULL)
   {
  ANTLR3_FREE(ctx);
  return  NULL;
   }
<if(memoize)>
<if(grammar.grammarIsRoot)>
   // Create a LIST for recording rule memos.
   //
   ctx->pLexer->rec->ruleMemo    = antlr3IntTrieNew(15); /* 16 bit depth is enough for 32768 rules! */
<endif>
<endif>

   /* Install the implementation of our <name> interface
    */
   <rules:{r | <if(!r.ruleDescriptor.isSynPred)>ctx->m<r.ruleDescriptor.name> =
m<r.ruleDescriptor.name>;<endif>}; separator="\n";>

   /** When the nextToken() call is made to this lexer's pANTLR3_TOKEN_SOURCE
     *  it will call mTokens() in this generated code, and will pass it the ctx
    * pointer of this lexer, not the context of the base lexer, so store that now.
    */
   ctx->pLexer->ctx     = ctx;

   /**Install the token matching function
    */
   ctx->pLexer->mTokens = (void (*) (void *))(mTokens);

   ctx->getGrammarFileName = getGrammarFileName;
   ctx->free  = <name>Free;
```

```
<if(grammar.directDelegates)>
 // Initialize the lexers that we are going to delegate some
 // functions to.
 //
 <grammar.directDelegates:
        {g|ctx-><g:delegateName()> = <g.recognizerName>NewSSD(instream, ctx->pLexer->rec->state,
ctx<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
<endif>
<if(grammar.delegators)>
 // Install the pointers back to lexers that will delegate us to perform certain functions
 // for them.
 //
 <grammar.delegators:
        {g|ctx-><g:delegateName()>   = <g:delegateName()>;}; separator="\n">
<endif>
<if(filterMode)>
   /*
 We have filter mode turned on, so install the filtering nextToken function
    */
   ctx->pLexer->rec->state->tokSource->nextToken = <name>NextToken;
<endif>
  <actions.lexer.apifuncs>

   /* Return the newly built lexer to the caller
    */
   return  ctx;
}
<if(cyclicDFAs)>

/* =========================================================================
* DFA tables for the lexer
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =========================================================================
* End of DFA tables for the lexer
*/
<endif>

/* =========================================================================
* Functions to match the lexer grammar defined tokens from the input stream
*/

<rules; separator="\n\n">

/* =========================================================================
* Lexer matching rules end.
* =========================================================================
*/
<if(synpreds)>

/* =========================================================================
 * Lexer syntactic predicates
*/
<synpreds:{p | <lexerSynpred(predname=p)>}>
/* =========================================================================
* Lexer syntactic predicates end.
* =========================================================================
*/
<endif>

/* End of Lexer code
* ================================================
* ================================================
*/

>>



filteringNextToken() ::= <<
/** An override of the lexer's nextToken() method that backtracks over mTokens() looking
*  for matches in lexer filterMode.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  BACKTRACKING needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at BACKTRACKING==1.
*/
static pANTLR3_COMMON_TOKEN
<name>NextToken(pANTLR3_TOKEN_SOURCE toksource)
{
   pANTLR3_LEXER   lexer;
 pANTLR3_RECOGNIZER_SHARED_STATE state;

   lexer
   = (pANTLR3_LEXER)(toksource->super);
   state = lexer->rec->state;

   /* Get rid of any previous token (token factory takes care of
    * any deallocation when this token is finally used up.
    */
   state  ->token     = NULL;
   state  ->error     = ANTLR3_FALSE;     /* Start out without an exception */
   state  ->failed    = ANTLR3_FALSE;

   /* Record the start of the token in our input stream.
    */
   state->tokenStartCharIndex   = lexer->input->istream->index(lexer->input->istream);
```

```
  state->tokenStartCharPositionInLine = lexer->input->getCharPositionInLine(lexer->input);
  state->tokenStartLine    = lexer->input->getLine(lexer->input);
  state->text       = NULL;

  /* Now call the matching rules and see if we can generate a new token
   */
  for (;;)
  {
 if  (lexer->input->istream->_LA(lexer->input->istream, 1) == ANTLR3_CHARSTREAM_EOF)
 {
  /* Reached the end of the stream, nothing more to do.
   */
  pANTLR3_COMMON_TOKEN    teof = &(toksource->eofToken);

  teof->setStartIndex
(teof, lexer->getCharIndex(lexer));
  teof->setStopIndex  (teof, lexer->getCharIndex(lexer));
  teof->setLine  (teof, lexer->getLine(lexer));
  return  teof;
 }

 state->token  = NULL;
 state->error  = ANTLR3_FALSE;     /* Start out without an exception */

 {
  ANTLR3_MARKER   m;

  m      = lexer->input->istream->mark(lexer->input->istream);
  state->backtracking  = 1;    /* No exceptions */
  state->failed   = ANTLR3_FALSE;

  /* Call the generated lexer, see if it can get a new token together.
   */
  lexer->mTokens(lexer->ctx);
    state->backtracking = 0;

    <! mTokens backtracks with synpred at BACKTRACKING==2
   and we set the synpredgate to allow actions at level 1. !>

  if (state->failed == ANTLR3_TRUE)
  {
   lexer->input->istream->rewind(lexer->input->istream, m);
   lexer->input->istream->consume(lexer->input->istream); <! advance one char and try again !>
  }
  else
  {
   lexer->emit(lexer);     /*
 Assemble the token and emit it to the stream */
    return state->token;
   }
  }
   }
}
>>

actionGate() ::= "BACKTRACKING==0"

filteringActionGate() ::= "BACKTRACKING==1"

/** How to generate a parser */
genericParser(  grammar,
    name,
               scopes,
               tokens,
               tokenNames,
               rules,
               numRules,
               bitsets,
               inputStreamType,
               superClass,
               ASTLabelType="pANTLR3_BASE_TREE",
               labelType,
    members,
    rewriteElementType, filterMode
             ) ::= <<



<if(grammar.grammarIsRoot)>
/** \brief Table of all token names in symbolic order, mainly used for
*         error reporting.
*/
pANTLR3_UINT8   <name>TokenNames[<length(tokenNames)>+4]
    = {
       (pANTLR3_UINT8) "\<invalid>",       /* String to print to indicate an invalid token */
       (pANTLR3_UINT8) "\<EOR>",
       (pANTLR3_UINT8)
 "\<DOWN>",
       (pANTLR3_UINT8) "\<UP>",
       <tokenNames:{(pANTLR3_UINT8) <it>}; separator=",\n">
      };
<endif>

   <@members>

   <@end>
```

```
<rules:{r |<ruleAttributeScopeFuncMacro(scope=r.ruleDescriptor.ruleScope)>}>
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeFuncMacro(scope=it)><endif>}>

// Forward declare the locally static matching functions we have generated.
//
<rules:{r | static <headerReturnType(ruleDescriptor=r.ruleDescriptor)> <r.ruleDescriptor.name>    (p<name>
ctx<if(r.ruleDescriptor.parameterScope)>, <endif><r.ruleDescriptor.parameterScope:parameterScope(scope=it)>);};
separator="\n";>
static void <name>Free(p<name> ctx);
<if(!LEXER)>
<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
<if(recognizer.grammar.delegatedRules)>
// Delegated rules
//
<recognizer.grammar.delegatedRules:{ruleDescriptor|static <headerReturnType(ruleDescriptor)>
<ruleDescriptor.name>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
 <endif><ruleDescriptor.parameterScope:parameterScope(scope=it)>);}; separator="\n";>

<endif>
<endif>

/* For use in tree output where we are accumulating rule labels via label += ruleRef
* we need a function that knows how to free a return scope when the list is destroyed.
* We cannot just use ANTLR3_FREE because in debug tracking mode, this is a macro.
*/
static void ANTLR3_CDECL freeScope(void * scope)
{
   ANTLR3_FREE(scope);
}

/** \brief Name of the grammar file that generated this code
*/
static const char fileName[] = "<fileName>";

/** \brief Return the name of the grammar file that generated this code.
*/
static const char * getGrammarFileName()
{
 return fileName;
}
/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
ANTLR3_API p<name>
```

```
<name>New   (<inputStreamType> instream<grammar.delegators:{g|, p<g.recognizerName>
 <g:delegateName()>}>)
{
 // See if we can create a new parser with the standard constructor
 //
 return <name>NewSSD(instream, NULL<grammar.delegators:{g|, <g:delegateName()>}>);
}

/** \brief Create a new <name> parser and return a context for it.
*
* \param[in] instream Pointer to an input stream interface.
*
* \return Pointer to new parser context upon success.
*/
ANTLR3_API p<name>
<name>NewSSD   (<inputStreamType> instream, pANTLR3_RECOGNIZER_SHARED_STATE
state<grammar.delegators:{g|, p<g.recognizerName> <g:delegateName()>}>)
{
   p<name> ctx;     /* Context structure we will build and return   */

   ctx = (p<name>) ANTLR3_CALLOC(1, sizeof(<name>));

   if (ctx == NULL)
   {
  // Failed to allocate memory for parser context
  //
       return  NULL;
   }

   /* -------------------------------------------------------------------
    * Memory for basic structure is allocated, now to fill in
    * the base ANTLR3 structures. We initialize the
 function pointers
    * for the standard ANTLR3 parser function set, but upon return
    * from here, the programmer may set the pointers to provide custom
    * implementations of each function.
    *
    * We don't use the macros defined in <name>.h here, in order that you can get a sense
    * of what goes where.
    */

<if(PARSER)>
   /* Create a base parser/recognizer, using the supplied token stream
    */
   ctx->pParser     = antlr3ParserNewStream(ANTLR3_SIZE_HINT, instream->tstream, state);
<endif>
<if(TREE_PARSER)>
   /* Create a base Tree parser/recognizer, using the supplied tree node stream
```

```
     */
    ctx->pTreeParser  = antlr3TreeParserNewStream(ANTLR3_SIZE_HINT, instream, state);
<endif>

    /* Install the implementation of our <name> interface
     */
    <rules:{r | ctx-><r.ruleDescriptor.name> = <r.ruleDescriptor.name>;}; separator="\n";>
<if(grammar.delegatedRules)>
 // Install the delegated methods so that they appear to be a part of this
 // parser
 //
     <grammar.delegatedRules:{ruleDescriptor | ctx-><ruleDescriptor.name> = <ruleDescriptor.name>;};
separator="\n";>
<endif>

    ctx->free   = <name>Free;
    ctx->getGrammarFileName = getGrammarFileName;

    /* Install the scope pushing methods.
     */
    <rules: {r |<if(r.ruleDescriptor.ruleScope)>
<ruleAttributeScope(scope=r.ruleDescriptor.ruleScope)><\n>
<endif>}>
    <recognizer.scopes:{<if(it.isDynamicGlobalScope)>
<globalAttributeScope(scope=it)><\n>
<endif>}>
    <@apifuncs>

    <@end>
<if(grammar.directDelegates)>
 // Initialize the parsers that we are going to delegate some
 // functions to.
 //
 <grammar.directDelegates:
         {g|ctx-><g:delegateName()> = <g.recognizerName>NewSSD(instream, PSRSTATE,
ctx<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
<endif>
<if(grammar.delegators)>
 // Install the pointers back to parsers that will delegate us to perform certain functions
 // for them.
 //
 <grammar.delegators:
         {g|ctx-><g:delegateName()>   =
 <g:delegateName()>;}; separator="\n">
<endif>
    <actions.parser.apifuncs>
    <actions.treeparser.apifuncs>
<if(memoize)>
```

```
<if(grammar.grammarIsRoot)>
   /* Create a LIST for recording rule memos.
    */
    RULEMEMO    = antlr3IntTrieNew(15); /* 16 bit depth is enough for 32768 rules! */<\n>
<endif>
<endif>
   /* Install the token table
    */
   PSRSTATE->tokenNames   = <grammar.composite.rootGrammar.recognizerName>TokenNames;

   <@debugStuff()>

   /* Return the newly built parser to the caller
    */
   return  ctx;
}

/** Free the parser resources
*/
static void
<name>Free(p<name> ctx)
{
   /* Free any scope memory
    */
   <rules: {r
|<if(r.ruleDescriptor.ruleScope)><ruleAttributeScopeFree(scope=r.ruleDescriptor.ruleScope)><\n><endif>}>
   <recognizer.scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeFree(scope=it)><\n><endif>}>

   <@cleanup>
   <@end>
<if(grammar.directDelegates)>
 // Free the parsers that we delegated to
 //
 functions to.NULL the state so we only free it once.
 //
 <grammar.directDelegates:
      {g| ctx-><g:delegateName()>-><if(TREE_PARSER)>pTreeParser<else>pParser<endif>->rec->state = NULL;
      ctx-><g:delegateName()>->free(ctx-><g:delegateName()>);}; separator="\n">
<endif>
<if(memoize)>
<if(grammar.grammarIsRoot)>
 if (RULEMEMO != NULL)
 {
  RULEMEMO->free(RULEMEMO);
  RULEMEMO = NULL;
 }
<endif>
<endif>
```

```
 // Free this parser
 //
<if(TREE_PARSER)>
   ctx->pTreeParser->free(ctx->pTreeParser);<\n>
<else>
   ctx->pParser->free(ctx->pParser);<\n>
<endif>
   ANTLR3_FREE(ctx);

   /* Everything is released, so we can return
    */
   return;
}

/** Return token names used by this <grammarType()>
*
* The returned pointer is used as an index into the token names table (using the token
* number as the index).
*
* \return Pointer to first char * in the table.
*/
static pANTLR3_UINT8    *getTokenNames()
{
       return <grammar.composite.rootGrammar.recognizerName>TokenNames;

}

   <members>

/* Declare the bitsets
*/
<bitsets:bitsetDeclare(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                 words64=it.bits)>


<if(cyclicDFAs)>

/* =========================================================================
* DFA tables for the parser
*/
<cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
/* =========================================================================
* End of DFA tables for the parser
*/
<endif>

/* ==============================================
* Parsing rules
```

```
*/
<rules; separator="\n\n">
<if(grammar.delegatedRules)>
 // Delegated methods that appear to be a part of this
 // parser
 //
<grammar.delegatedRules:{ruleDescriptor|
   <returnType()> <ruleDescriptor.name>(p<name> ctx<if(ruleDescriptor.parameterScope.attributes)>,
<endif><ruleDescriptor.parameterScope:parameterScope(scope=it)>)
   \{
       <if(ruleDescriptor.hasReturnValue)>return <endif>ctx-><ruleDescriptor.grammar:delegateName()>-
><ruleDescriptor.name>(ctx-
><ruleDescriptor.grammar:delegateName()><if(ruleDescriptor.parameterScope.attributes)>,
 <endif><ruleDescriptor.parameterScope.attributes:{a|<a.name>}; separator=", ">);
 \}}; separator="\n">

<endif>
/* End of parsing rules
* ==============================================
*/

/* ==============================================
* Syntactic predicates
*/
<synpreds:{p | <synpred(predname=p)>}>
/* End of syntactic predicates
* ==============================================
*/





>>

parser( grammar,
  name,
  scopes,
  tokens,
  tokenNames,
  rules,
  numRules,
  bitsets,
  ASTLabelType,
  superClass="Parser",
  labelType="pANTLR3_COMMON_TOKEN",
  members={<actions.parser.members>}
  ) ::= <<
```

```
<genericParser(inputStreamType="pANTLR3_COMMON_TOKEN_STREAM", rewriteElementType="TOKEN",
...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser( grammar,
   name,
   scopes,
   tokens,

   tokenNames,
   globalAction,
   rules,
   numRules,
   bitsets,
   labelType={<ASTLabelType>},
   ASTLabelType="pANTLR3_BASE_TREE",
   superClass="TreeParser",
   members={<actions.treeparser.members>}, filterMode
   ) ::= <<
<genericParser(inputStreamType="pANTLR3_COMMON_TREE_NODE_STREAM",
rewriteElementType="NODE", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
static void <ruleName>_fragment(p<name> ctx <ruleDescriptor.parameterScope:parameterScope(scope=it)>)
{
 <ruleLabelDefs()>
 <ruleLabelInitializations()>
<if(trace)>
    ANTLR3_PRINTF("enter <ruleName> %d failed = %d, backtracking =
%d\\n",LT(1),failed,BACKTRACKING);
   <block>
   ANTLR3_PRINTF("exit <ruleName> %d, failed = %d, backtracking = %d\\n",LT(1),failed,BACKTRACKING);

<else>
   <block>
```

```
<endif>
<ruleCleanUp()>
}
// $ANTLR end <ruleName>
>>

synpred(predname) ::= <<
static ANTLR3_BOOLEAN <predname>(p<name> ctx)
{
   ANTLR3_MARKER   start;
   ANTLR3_BOOLEAN  success;

   BACKTRACKING++;
   <@start()>
   start = MARK();
   <predname>_fragment(ctx);     // can never throw exception
   success = !(FAILEDFLAG);
   REWIND(start);
   <@stop()>
   BACKTRACKING--;
   FAILEDFLAG = ANTLR3_FALSE;
   return success;
}<\n>
>>

lexerSynpred(predname) ::= <<
<synpred(predname)>
>>

ruleMemoization(rname) ::= <<
<if(memoize)>
if ( (BACKTRACKING>0) && (HAVEPARSEDRULE(<ruleDescriptor.index>)) )
{
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!ruleDescriptor.isSynPred)>
 retval.start = 0;
 <scopeClean()><\n>
<endif>
<endif>
    return <ruleReturnValue()>;
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
if  (HASEXCEPTION())
{
   goto rule<ruleDescriptor.name>Ex;
}
<if(backtracking)>
if (HASFAILED())
{
   <scopeClean()>
   return <ruleReturnValue()>;
}
<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if (BACKTRACKING>0)
{
   FAILEDFLAG = <true()>;
   <scopeClean()>
   return <ruleReturnValue()>;
}
<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
/**
* $ANTLR start <ruleName>
* <fileName>:<description>
*/
static <returnType()>
<ruleName>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope(scope=it)>)
{
   <if(trace)>ANTLR3_PRINTF("enter
 <ruleName> %s failed=%d, backtracking=%d\n", LT(1), BACKTRACKING);<endif>
   <ruleDeclarations()>
   <ruleDescriptor.actions.declarations>
   <ruleLabelDefs()>
   <ruleInitializations()>
   <ruleDescriptor.actions.init>
   <ruleMemoization(rname=ruleName)>
   <ruleLabelInitializations()>
   <@preamble()>
   {
       <block>
   }
   <ruleCleanUp()>
<if(exceptions)>
   if (HASEXCEPTION())
   {
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
   }
   else
   {
 <(ruleDescriptor.actions.after):execAction()>
   }
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   if (HASEXCEPTION())
   {
       PREPORTERROR();
       PRECOVER();
       <@setErrorReturnValue()>
   }
<if(ruleDescriptor.actions.after)>
   else
   {
 <(ruleDescriptor.actions.after):execAction()>
   }<\n>
<endif>
<endif>
<endif>
<endif>
   <if(trace)>System.out.println("exit <ruleName> "+LT(1)+" failed="+failed+"
 backtracking="+BACKTRACKING);<endif>
   <memoize()>
   <finally>
   <@postamble()>
   return <ruleReturnValue()>;
}
/* $ANTLR end <ruleName> */
>>

catch(decl,action) ::= <<
/* catch(decl,action)
*/
if  ((HASEXCEPTION()) && (EXCEPTION->type == <e.decl>) )
{
   <e.action>
}
```

```
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval;<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
ANTLR3_UINT32 <ruleDescriptor.name>_StartIndex;
<endif>
>>

ruleInitializations() ::= <<
/* Initialize rule variables
*/
<if(memoize)>
<ruleDescriptor.name>_StartIndex = INDEX();<\n>
<endif>
<ruleDescriptor.useScopes:{<scopeTop(sname=it)> = <scopePush(sname=it)>;}; separator="\n">
<ruleDescriptor.ruleScope:{<scopeTop(sname=it.name)> = <scopePush(sname=it.name)>;}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
    :{<labelType>    <it.label.text>;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{pANTLR3_VECTOR    list_<it.label.text>;}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
   :ruleLabelDef(label=it); separator="\n"
>
>>

ruleLabelInitializations() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
   :{<it.label.text>       = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{list_<it.label.text>     = NULL;}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
   :ruleLabelInitVal(label=it); separator="\n"
>
<if(ruleDescriptor.hasMultipleReturnValues)>
```

```
<if(!ruleDescriptor.isSynPred)>
retval.start = LT(1); retval.stop = retval.start;<\n>
<endif>
<endif>
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<labelType> <it.label.text>;};
 separator="\n"
>
<ruleDescriptor.charLabels:{ANTLR3_UINT32 <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{pANTLR3_INT_TRIE list_<it.label.text>;}; separator="\n"
>
>>

lexerRuleLabelInit() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{list_<it.label.text> = antlr3IntTrieNew(31);}; separator="\n"
>
>>

lexerRuleLabelFree() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<it.label.text> = NULL;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleListLabels]
   :{list_<it.label.text>->free(list_<it.label.text>);}; separator="\n"
>
>>

ruleReturnValue() ::= <<
```

```
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

memoize()
 ::= <<
<if(memoize)>
<if(backtracking)>
if ( BACKTRACKING>0 ) { MEMOIZE(<ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

ruleCleanUp() ::= <<

// This is where rules clean up and exit
//
goto rule<ruleDescriptor.name>Ex; /* Prevent compiler warnings */
rule<ruleDescriptor.name>Ex: ;
<scopeClean()>
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
<if(!ruleDescriptor.isSynPred)>
retval.stop = LT(-1);<\n>
<endif>
<endif>
<endif>
>>

scopeClean() ::= <<
<ruleDescriptor.useScopes:{<scopePop(sname=it)>}; separator="\n">
<ruleDescriptor.ruleScope:{<scopePop(sname=it.name)>}; separator="\n">

>>
/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules, which do not produce tokens.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize)
 ::= <<
//   Comes from: <block.description>
/** \brief Lexer rule generated by ANTLR3
```

```
*
* $ANTLR start <ruleName>
*
* Looks to match the characters the constitute the token <ruleName>
* from the attached input stream.
*
*
* \remark
*  - lexer->error == ANTLR3_TRUE if an exception was thrown.
*/
static ANTLR3_INLINE
void m<ruleName>(p<name> ctx<if(ruleDescriptor.parameterScope)>,
<endif><ruleDescriptor.parameterScope:parameterScope(scope=it)>)
{
 ANTLR3_UINT32 _type;
   <ruleDeclarations()>
   <ruleDescriptor.actions.declarations>
   <lexerRuleLabelDefs()>
   <if(trace)>System.out.println("enter <ruleName> '"+(char)LA(1)+"'
line="+GETLINE()+":"+GETCHARPOSITIONINLINE()+" failed="+failed+"
backtracking="+BACKTRACKING);<endif>

<if(nakedBlock)>
   <ruleMemoization(rname=ruleName)>
   <lexerRuleLabelInit()>
   <ruleDescriptor.actions.init>

   <block><\n>
<else>
   <ruleMemoization(rname=ruleName)>
   <lexerRuleLabelInit()>
   _type     = <ruleName>;


   <ruleDescriptor.actions.init>

   <block>
 LEXSTATE->type = _type;
<endif>
   <if(trace)> ANTLR3_FPRINTF(stderr, "exit <ruleName> '%c' line=%d:%d failed = %d, backtracking
=%d\n",LA(1),GETLINE(),GETCHARPOSITIONINLINE(),failed,BACKTRACKING);<endif>
   <ruleCleanUp()>
   <lexerRuleLabelFree()>
   <(ruleDescriptor.actions.after):execAction()>
   <memoize>
}
// $ANTLR end <ruleName>
>>
```

```
/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
/** This is the entry point in to the lexer from an object that
*  wants to generate the next token, such as a pCOMMON_TOKEN_STREAM
*/
static void
mTokens(p<name> ctx)
{
   <block><\n>

   goto ruleTokensEx; /* Prevent compiler warnings */
ruleTokensEx: ;
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<

// <fileName>:<description>
{
   int alt<decisionNumber>=<maxAlt>;
   <decls>
   <@predecision()>
   <decision>
   <@postdecision()>
   <@prebranch()>
   switch (alt<decisionNumber>)
   {
 <alts:altSwitchCase()>
   }
   <@postbranch()>
}
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
{
   //  <fileName>:<description>

   ANTLR3_UINT32 alt<decisionNumber>;

   alt<decisionNumber>=<maxAlt>;
   <decls>
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
   {
 <alts:altSwitchCase()>
   }
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description)
 ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
{
   int cnt<decisionNumber>=0;
   <decls>
   <@preloop()>

   for (;;)
   {
       int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch (alt<decisionNumber>)
 {
     <alts:altSwitchCase()>
```

```
    default:

  if ( cnt<decisionNumber> >= 1 )
  {
      goto loop<decisionNumber>;
  }
  <ruleBacktrackFailure()>
  <earlyExitEx()>
  <@earlyExitException()>
  goto rule<ruleDescriptor.name>Ex;
 }
 cnt<decisionNumber>++;
   }
   loop<decisionNumber>: ; /* Jump to here if this rule does not match */
   <@postloop()>
}
>>

earlyExitEx() ::= <<
/* mismatchedSetEx()
*/
CONSTRUCTEX();
EXCEPTION->type = ANTLR3_EARLY_EXIT_EXCEPTION;
EXCEPTION->name
 = (void *)ANTLR3_EARLY_EXIT_NAME;
<\n>
>>
positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<

// <fileName>:<description>
<decls>

<@preloop()>
for (;;)
{
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
   {
 <alts:altSwitchCase()>
 default:
    goto loop<decisionNumber>; /* break out of the loop */
    break;
  }
}
loop<decisionNumber>: ; /* Jump out to here if this rule does not match */
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by antlr before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts
 the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i>:
  <@prealt()>
  <it>
  break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
  <@declarations()>
  <@initializations()>
  <elements:element()>
  <rew>
  <@cleanup()>
}
>>

// E L E M E N T S
/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

/** Dump the elements one per line */
element() ::= <<
```

```
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label> = (<labelType>)<endif> MATCHT(<token>,
&FOLLOW_<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == NULL)
{
   list_<label>=ctx->vectors->newVector(ctx->vectors);
}
list_<label>->add(list_<label>, <elem>, NULL);
>>



/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHC(<char>);
<checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHRANGE(<a>, <b>);
<checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
```

```
<if(LEXER)>
<label>= LA(1);<\n>
<else>
<label>=(<labelType>)LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
   CONSUME();
   <postmatchCode>
<if(!LEXER)>
   PERRORRECOVERY=ANTLR3_FALSE;
<endif>
   <if(backtracking)>FAILEDFLAG=ANTLR3_FALSE;<\n><endif>
}
else
{
   <ruleBacktrackFailure()>
   <mismatchedSetEx()>
   <@mismatchedSetException()>
<if(LEXER)>
    LRECOVER();
<else>
   RECOVERFROMMISMATCHEDSET(&FOLLOW_set_in_<ruleName><elementIndex>);
<endif>
   goto rule<ruleDescriptor.name>Ex;
}<\n>
>>

mismatchedSetEx() ::= <<
CONSTRUCTEX();
EXCEPTION->type         = ANTLR3_MISMATCHED_SET_EXCEPTION;
EXCEPTION->name         = (void *)ANTLR3_MISMATCHED_SET_NAME;
<if(PARSER)>
EXCEPTION->expectingSet = &FOLLOW_set_in_<ruleName><elementIndex>;
<endif>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
```

```
<label>Start = GETCHARINDEX();
MATCHS(<string>);
<checkRuleBacktrackFailure()>
<label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
<label>->setType(<label>, ANTLR3_TOKEN_INVALID);
<label>->setStartIndex(<label>, <label>Start);
<label>->setStopIndex(<label>, GETCHARINDEX()-1);
<label>->input = INPUT->tnstream->istream;
<else>
MATCHS(<string>);

<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label>=(<labelType>)LT(1);<\n>
<endif>
MATCHANYT();
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = LA(1);<\n>
<endif>
MATCHANY();
<checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values. The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
```

```
ruleRef(rule,label,elementIndex,args,scope) ::= <<
FOLLOWPUSH(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)>ctx-><scope:delegateName()>-><endif><rule.name>(ctx<if(scope)>-
><scope:delegateName()><endif><if(args)>,
 <args; separator=", "><endif>);<\n>
FOLLOWPOP();
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
/* <description> */
<if(label)>
{
   ANTLR3_MARKER <label>Start<elementIndex> = GETCHARINDEX();
   <if(scope)>ctx-><scope:delegateName()>-><endif>m<rule.name>(ctx<if(scope)>-
><scope:delegateName()><endif> <if(args)>, <endif><args; separator=", ">);
   <checkRuleBacktrackFailure()>
   <label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
   <label>->setType(<label>, ANTLR3_TOKEN_INVALID);
   <label>->setStartIndex(<label>, <label>Start<elementIndex>);
    <label>->setStopIndex(<label>, GETCHARINDEX()-1);
   <label>->input = INPUT;
}
<else>
<if(scope)>ctx-><scope:delegateName()>-><endif>m<rule.name>(ctx<if(scope)>-
><scope:delegateName()><endif> <if(args)>, <endif><args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
```

```
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
{
   ANTLR3_UINT32 <label>Start<elementIndex>;
   <labelType> <label>;
   <label>Start<elementIndex> = GETCHARINDEX();
   MATCHC(ANTLR3_CHARSTREAM_EOF);
   <checkRuleBacktrackFailure()>
   <label> = LEXSTATE->tokFactory->newToken(LEXSTATE->tokFactory);
   <label>->setType(<label>, ANTLR3_TOKEN_EOF);
   <label>->setStartIndex(<label>, <label>Start<elementIndex>);
   <label>->setStopIndex(<label>, GETCHARINDEX()-1);
   <label>->input = INPUT->tnstream->istream;
}
<else>
    MATCHC(ANTLR3_CHARSTREAM_EOF);
   <checkRuleBacktrackFailure()>
   <endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( LA(1)==ANTLR3_TOKEN_DOWN ) {
   MATCHT(ANTLR3_TOKEN_DOWN, NULL);
   <checkRuleBacktrackFailure()>
   <children:element()>
   MATCHT(ANTLR3_TOKEN_UP, NULL);
   <checkRuleBacktrackFailure()>
}
<else>
MATCHT(ANTLR3_TOKEN_DOWN, NULL);
<checkRuleBacktrackFailure()>
<children:element()>
MATCHT(ANTLR3_TOKEN_UP, NULL);
<checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) )
{
   <ruleBacktrackFailure()>
```

```
  <newFPE(...)>
}
>>

newFPE() ::= <<
  CONSTRUCTEX();
  EXCEPTION->type
      = ANTLR3_FAILED_PREDICATE_EXCEPTION;
  EXCEPTION->message      = (void *)"<description>";
  EXCEPTION->ruleName  = (void *)"<ruleName>";
  <\n>
>>

// Fixed DFA (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<

{
  int LA<decisionNumber>_<stateNumber> = LA(<k>);
  <edges; separator="\nelse ">
  else
  {
<if(eotPredictsAlt)>
      alt<decisionNumber>=<eotPredictsAlt>;
<else>
      <ruleBacktrackFailure()>

      <newNVException()>
      goto rule<ruleDescriptor.name>Ex;

<endif>
  }
}
>>

newNVException() ::= <<
CONSTRUCTEX();
EXCEPTION->type         = ANTLR3_NO_VIABLE_ALT_EXCEPTION;
EXCEPTION->message      = (void *)"<description>";
EXCEPTION->decisionNum  = <decisionNumber>;
EXCEPTION->state        = <stateNumber>;
<@noViableAltException()>
<\n>
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster,
 smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
   int LA<decisionNumber>_<stateNumber> = LA(<k>);
   <edges; separator="\nelse ">
}
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/

dfaLoopbackStateDecls()::= <<
ANTLR3_UINT32   LA<decisionNumber>_<stateNumber>;
>>
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
  /* dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
   */
   int LA<decisionNumber>_<stateNumber> = LA(<k>);
   <edges; separator="\nelse "><\n>
   <if(eotPredictsAlt)>
    <if(!edges)>
 alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
 <else>
   else
   {
 alt<decisionNumber>=<eotPredictsAlt>;
   }<\n>
   <endif>
   <endif>
}
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>)
```

```
{
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k>) )
{
<edges; separator="\n">

default:
<if(eotPredictsAlt)>
    alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   <newNVException()>
   goto rule<ruleDescriptor.name>Ex;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k>) )
{
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
   alt<decisionNumber>=<eotPredictsAlt>;
   break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n">
 {
  <targetState>
 }
   break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(ctx,
 RECOGNIZER, ISTREAM, &cdfa<decisionNumber>);
<checkRuleBacktrackFailure()>
>>

/* Dump DFA tables as static initialized arrays of shorts(16 bits)/characters(8 bits)
* which are then used to statically initialize the dfa structure, which means that there
* is no runtime initialization whatsoever, other than anything the C compiler might
* need to generate. In general the C compiler will lay out memory such that there is no
* runtime code required.
*/
cyclicDFA(dfa) ::= <<
/** Static dfa state tables for Cyclic dfa:
*    <dfa.description>
*/
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_eot[<dfa.numberOfStates>] =
   {
 <dfa.eot; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_eof[<dfa.numberOfStates>] =
   {
 <dfa.eof; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_min[<dfa.numberOfStates>] =
   {
 <dfa.min; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32
 dfa<dfa.decisionNumber>_max[<dfa.numberOfStates>] =
   {
 <dfa.max; wrap="\n", separator=", ", null="-1">
   };
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_accept[<dfa.numberOfStates>] =
   {
 <dfa.accept; wrap="\n", separator=", ", null="-1">
   };
```

```
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_special[<dfa.numberOfStates>] =
   {
 <dfa.special; wrap="\n", separator=", ", null="-1">
   };

/** Used when there is no transition table entry for a particular state */
#define dfa<dfa.decisionNumber>_T_empty     NULL

<dfa.edgeTransitionClassMap.keys:{ table |
static const ANTLR3_INT32 dfa<dfa.decisionNumber>_T<i0>[] =
   {
 <table; separator=", ", wrap="\n", null="-1">
   };}; null = "">

/* Transition tables are a table of sub tables, with some tables
* reused for efficiency.
*/
static const ANTLR3_INT32 * const dfa<dfa.decisionNumber>_transitions[] =
{
   <dfa.transitionEdgeTables:{xref|dfa<dfa.decisionNumber>_T<xref>}; separator=", ", wrap="\n", null="_empty">
};

<if(dfa.specialStateSTs)>
static
 ANTLR3_INT32 dfa<dfa.decisionNumber>_sst(p<name> ctx, pANTLR3_BASE_RECOGNIZER recognizer,
pANTLR3_INT_STREAM is, pANTLR3_CYCLIC_DFA dfa, ANTLR3_INT32 s)
{
   ANTLR3_INT32    _s;

   _s     = s;
   switch  (s)
   {
   <dfa.specialStateSTs:{state |
   case <i0>:

 <state>}; separator="\n">
   }
<if(backtracking)>
   if (BACKTRACKING > 0)
   {
 FAILEDFLAG = ANTLR3_TRUE;
 return -1;
   }
<endif>

   CONSTRUCTEX();
   EXCEPTION->type         = ANTLR3_NO_VIABLE_ALT_EXCEPTION;
   EXCEPTION->message      = (void *)"<dfa.description>";
   EXCEPTION->decisionNum  = <dfa.decisionNumber>;
   EXCEPTION->state        = _s;
   <@noViableAltException()>
   return -1;
}
<endif>

<@errorMethod()>

/* Declare tracking structure for Cyclic DFA <dfa.decisionNumber>
*/
static
ANTLR3_CYCLIC_DFA cdfa<dfa.decisionNumber>
   = {
     <dfa.decisionNumber>,       /* Decision number of this dfa      */
     /* Which decision this represents:   */

    (const pANTLR3_UCHAR)"<dfa.description>",
<if(dfa.specialStateSTs)>
     (CDFA_SPECIAL_FUNC) dfa<dfa.decisionNumber>_sst,
<else>
     (CDFA_SPECIAL_FUNC) antlr3dfaspecialStateTransition, /* Default special state transition function */
<endif>

     antlr3dfaspecialTransition,  /* DFA specialTransition is currently just a default function in the runtime */
     antlr3dfapredict,   /* DFA simulator function is in the runtime */
     dfa<dfa.decisionNumber>_eot,     /* EOT table       */
     dfa<dfa.decisionNumber>_eof,     /* EOF table       */
     dfa<dfa.decisionNumber>_min,     /* Minimum tokens for each state    */
     dfa<dfa.decisionNumber>_max,     /* Maximum tokens for each state    */
     dfa<dfa.decisionNumber>_accept, /* Accept table       */
     dfa<dfa.decisionNumber>_special, /* Special transition states     */
     dfa<dfa.decisionNumber>_transitions /* Table of transition tables     */

 };
/* End of Cyclic DFA <dfa.decisionNumber>
* ---------------------
 */
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
{
   ANTLR3_UINT32 LA<decisionNumber>_<stateNumber>;<\n>
   ANTLR3_MARKER index<decisionNumber>_<stateNumber>;<\n>
 LA<decisionNumber>_<stateNumber> = LA(1);<\n>
   <if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
   index<decisionNumber>_<stateNumber> = INDEX();<\n>
   REWINDLAST();<\n>
   <endif>
   s = -1;
   <edges; separator="\nelse ">
 <if(semPredState)> <! return input cursor to state before we rewound !>
 SEEK(index<decisionNumber>_<stateNumber>);<\n>
 <endif>
   if ( s>=0 )
   {
 return s;
   }
}
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>)
{
    s = <targetStateNumber>;
}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "( (<left>) && (<right>) )"

orPredicates(operands) ::= "((<first(operands)>)<rest(operands):{o | ||(<o>)}>)"

notPredicate(pred) ::= "!( <evalPredicate(...)> )"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>(ctx)"
```

```
lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
((LA<decisionNumber>_<stateNumber> \>= <lower>) && (LA<decisionNumber>_<stateNumber>
 \<= <upper>))
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "((LA(<k>) \>= <lower>)
&& (LA(<k>) \<= <upper>))"

setTest(ranges) ::= "<ranges; separator=\" || \">"

// A T T R I B U T E S

makeScopeSet() ::= <<
/* makeScopeSet()
*/
/** Definition of the <scope.name> scope variable tracking
*  structure. An instance of this structure is created by calling
*  <name>_<scope.name>Push().
*/
typedef struct  <scopeStruct(sname=scope.name,...)>_struct
{
   /** Function that the user may provide to be called when the
    *  scope is destroyed (so you can free pANTLR3_HASH_TABLES and so on)
    *
    * \param POinter to an instance of this typedef/struct
    */
   void    (ANTLR3_CDECL *free) (struct <scopeStruct(sname=scope.name,...)>_struct * frame);

   /* =============================================================================
    * Programmer defined variables...
    */
   <scope.attributes:{<it.decl>;}; separator="\n">

    /* End of programmer defined variables
    * =============================================================================
    */
}
   <scopeStruct(sname=scope.name,...)>, * <scopeType(sname=scope.name,...)>;

>>
```

```
globalAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

ruleAttributeScopeDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDecl(scope)
*/
<makeScopeSet(...)>
<endif>
>>

globalAttributeScopeFuncDecl(scope) ::=
<<
/* globalAttributeScopeFuncDecl(scope)
*/
<if(scope.attributes)>
/* -----------------------------------------------------------------------------
* Function declaration for creating a <name>_<scope.name> scope set
*/
static <scopeType(sname=scope.name,...)>   <scopePushName(sname=scope.name,...)>(p<name> ctx);
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope);
/* -----------------------------------------------------------------------------
 */

<endif>
>>

globalAttributeScopeFuncMacro(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeFuncMacro(scope)
*/
/** Function  for popping the top value from a <scopeStack(sname=scope.name)>
*/
void
<scopePopName(sname=scope.name,...)>(p<name> ctx)
{
   // First see if the user defined a function they want to be called when a
   // scope is popped/freed.
   //
 // If the user supplied the scope entries with a free function,then call it first
 //
   if (SCOPE_TOP(<scope.name>)->free != NULL)
 {
     SCOPE_TOP(<scope.name>)->free(SCOPE_TOP(<scope.name>));
 }

   // Now we decrement the scope's upper limit bound. We do not actually pop the scope as
   // we want to reuse scope entries if we do continuous push and pops. Most scopes don't
   // next too far so we don't want to keep freeing and allocating them
   //
   ctx-\><scopeStack(sname=scope.name,...)>_limit--;
   SCOPE_TOP(<scope.name>) = (<scopeType(sname=scope.name)>)(ctx-\><scopeStack(sname=scope.name,...)>-
\>get(ctx-\><scopeStack(sname=scope.name,...)>,
 ctx-\><scopeStack(sname=scope.name,...)>_limit - 1));
}
<endif>
>>

ruleAttributeScopeFuncDecl(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncDecl(scope)
*/
/* -----------------------------------------------------------------------------
* Function declarations for creating a <name>_<scope.name> scope set
*/
static <scopeType(sname=scope.name,...)>   <scopePushName(sname=scope.name,...)>(p<name> ctx);
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope);
/* ----------------------------------------------------------------------------- */

<endif>
>>

ruleAttributeScopeFuncMacro(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncMacro(scope)
*/
/** Function for popping the top value from a <scopeStack(sname=scope.name,...)>
*/
void
<scopePopName(sname=scope.name,...)>(p<name> ctx)
{
   // First see if the user defined a function they want to be called when a
   // scope
 is popped/freed.
   //
 // If the user supplied the scope entries with a free function,then call it first
 //
   if (SCOPE_TOP(<scope.name>)->free != NULL)
 {
     SCOPE_TOP(<scope.name>)->free(SCOPE_TOP(<scope.name>));
 }
```

```
   // Now we decrement the scope's upper limit bound. We do not actually pop the scope as
   // we want to reuse scope entries if we do continuous push and pops. Most scopes don't
   // next too far so we don't want to keep freeing and allocating them
   //
   ctx-\><scopeStack(sname=scope.name,...)>_limit--;
   SCOPE_TOP(<scope.name>) = (<scopeType(sname=scope.name)>)(ctx-\><scopeStack(sname=scope.name,...)>-
\>get(ctx-\><scopeStack(sname=scope.name,...)>, ctx-\><scopeStack(sname=scope.name,...)>_limit - 1));
}

<endif>
>>

globalAttributeScopeDef(scope) ::=
<<
/* globalAttributeScopeDef(scope)
*/
<if(scope.attributes)>
/** Pointer to the  <scope.name> stack for use by <scopePushName(sname=scope.name)>()
*  and <scopePopName(sname=scope.name,...)>()
 */
pANTLR3_STACK <scopeStack(sname=scope.name)>;
ANTLR3_UINT32 <scopeStack(sname=scope.name)>_limit;
/** Pointer to the top of the stack for the global scope <scopeStack(sname=scope.name)>
*/
<scopeType(sname=scope.name,...)>    (*<scopePushName(sname=scope.name,...)>)(struct <name>_Ctx_struct *
ctx);
<scopeType(sname=scope.name,...)>    <scopeTopDecl(sname=scope.name,...)>;

<endif>
>>

ruleAttributeScopeDef(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeDef(scope)
*/
/** Pointer to the  <scope.name> stack for use by <scopePushName(sname=scope.name)>()
*  and <scopePopName(sname=scope.name,...)>()
*/
pANTLR3_STACK <scopeStack(sname=scope.name,...)>;
ANTLR3_UINT32 <scopeStack(sname=scope.name,...)>_limit;
<scopeType(sname=scope.name,...)>   (*<scopePushName(sname=scope.name,...)>)(struct <name>_Ctx_struct *
ctx);
<scopeType(sname=scope.name,...)>   <scopeTopDecl(sname=scope.name,...)>;

<endif>
>>
```

```
globalAttributeScopeFuncs(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScopeFuncs(scope)
 */
<attributeFuncs(scope)>
<endif>
>>

ruleAttributeScopeFuncs(scope) ::= <<
<if(scope.attributes)>
/* ruleAttributeScopeFuncs(scope)
*/
<attributeFuncs(scope)>
<endif>
>>

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScope(scope)
*/
ctx-><scopePushName(sname=scope.name,...)>     = <scopePushName(sname=scope.name,...)>;
ctx-><scopeStack(sname=scope.name,...)>    = antlr3StackNew(0);
ctx-><scopeStack(sname=scope.name,...)>_limit    = 0;
<scopeTop(sname=scope.name,...)>      = NULL;
<endif>
>>

ruleAttributeScope(scope) ::=
<<
<if(scope.attributes)>
/* ruleAttributeScope(scope)
*/
ctx-><scopePushName(sname=scope.name,...)>     = <scopePushName(sname=scope.name,...)>;
ctx-><scopeStack(sname=scope.name,...)>    = antlr3StackNew(0);
ctx-><scopeStack(sname=scope.name,...)>_limit    = 0;
<scopeTop(sname=scope.name,...)>      = NULL;
<endif>
>>
globalAttributeScopeFree(scope) ::= <<
<if(scope.attributes)>
/* globalAttributeScope(scope)

*/
ctx-><scopeStack(sname=scope.name,...)>-\>free(ctx-><scopeStack(sname=scope.name,...)>);
<endif>
>>

ruleAttributeScopeFree(scope) ::=
<<
<if(scope.attributes)>
/* ruleAttributeScope(scope)
*/
ctx-><scopeStack(sname=scope.name,...)>-\>free(ctx-><scopeStack(sname=scope.name,...)>);
<endif>
>>

scopeTopDecl(sname) ::= <<
p<name>_<sname>Top
>>

scopeTop(sname) ::= <<
ctx-><scopeTopDecl(sname=sname,...)>
>>

scopePop(sname) ::= <<
<scopePopName(sname=sname,...)>(ctx);
>>

scopePush(sname) ::= <<
p<name>_<sname>Push(ctx)
>>

scopePopName(sname) ::= <<
p<name>_<sname>Pop
>>

scopePushName(sname) ::= <<
p<name>_<sname>Push
>>

scopeType(sname) ::= <<
p<name>_<sname>_SCOPE
>>

scopeStruct(sname) ::= <<
<name>_<sname>_SCOPE
>>

scopeStack(sname) ::= <<
p<name>_<sname>Stack
>>

attributeFuncs(scope) ::= <<
<if(scope.attributes)>
/* attributeFuncs(scope)
*/
```

```
static void ANTLR3_CDECL <scope.name>Free(<scopeType(sname=scope.name)> scope)
{
   ANTLR3_FREE(scope);
}

/** \brief Allocate
 initial memory for a <name> <scope.name> scope variable stack entry and
*         add it to the top of the stack.
*
* \remark
* By default the structure is freed with ANTLR_FREE(), but you can use the
* the \@init action to install a pointer to a custom free() routine by
* adding the code:
* \code
*   <scopeTop(sname=scope.name)>->free = myroutine;
* \endcode
*
* With lots of comments of course! The routine should be declared in
* \@members { } as:
* \code
*   void ANTLR3_CDECL myfunc( <scopeType(sname=scope.name)> ptr).
* \endcode
*
* It should perform any custom freeing stuff that you need (call ANTLR_FREE3, not free()
* NB: It should not free the pointer it is given, which is the scope stack entry itself
* and will be freed by the function that calls your custom free routine.
*
*/
static <scopeType(sname=scope.name)>
<scopePushName(sname=scope.name)>(p<name> ctx)
{
   /* Pointer used to create a new set of attributes
    */
   <scopeType(sname=scope.name)>
      newAttributes;

   /* Allocate the memory for a new structure if we need one.
    */
   if (ctx-\><scopeStack(sname=scope.name)>-\>size(ctx-\><scopeStack(sname=scope.name)>) \> ctx-
\><scopeStack(sname=scope.name)>_limit)
   {
       // The current limit value was less than the number of scopes available on the stack so
       // we can just reuse one. Our limit tracks the stack count, so the index of the entry we want
       // is one less than that, or conveniently, the current value of limit.
       //
       newAttributes = (<scopeType(sname=scope.name)>)ctx-><scopeStack(sname=scope.name)>->get(ctx-
><scopeStack(sname=scope.name)>, ctx-\><scopeStack(sname=scope.name)>_limit);
```

```
   }
   else
   {
       // Need a new allocation
       //
       newAttributes = (<scopeType(sname=scope.name)>)
ANTLR3_MALLOC(sizeof(<scopeStruct(sname=scope.name)>));
       if  (newAttributes != NULL)
       {
           /* Standard ANTLR3 library implementation
            */
            ctx-\><scopeStack(sname=scope.name)>-\>push(ctx-\><scopeStack(sname=scope.name)>, newAttributes,
(void (*)(void *))<scope.name>Free);
       }
   }

   // Blank out any previous free pointer, the user might or might install a new one.
   //
   newAttributes->free = NULL;

   // Indicate the position in the available stack that the current level is at
   //
   ctx-><scopeStack(sname=scope.name)>_limit++;

 /* Return value is the pointer to the new entry, which may be used locally
  * without de-referencing via the context.
    */
   return  newAttributes;
}<\n>

<endif>
>>
returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
ANTLR3_BOOLEAN
<endif>
```

```
>>

/** Generate the C type associated
 with a single or multiple return
*  value(s).
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>_<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "0".
*/
initValue(typeName) ::= <<
<cTypeInitMap.(typeName)>
>>

/** Define a rule label  */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text>;
#undef RETURN_TYPE_<label.label.text>
#define RETURN_TYPE_<label.label.text> <ruleLabelType(referencedRule=label.referencedRule)><\n>
>>
/**  Rule label default value */
ruleLabelInitVal(label) ::= <<
<if(label.referencedRule.hasSingleReturnValue)>
<label.label.text> = <initValue(label.referencedRule.singleValueReturnType)>;
<endif>
>>

ASTLabelType()
 ::= "<if(recognizer.ASTLabelType)><recognizer.ASTLabelType><else>pANTLR3_BASE_TREE<endif>"

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
```

```
*/
returnScope() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
typedef struct <ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>_struct
{
<if(!TREE_PARSER)>
  /** Generic return elements for ANTLR3 rules that are not in tree parsers or returning trees
   */
  pANTLR3_COMMON_TOKEN    start;
  pANTLR3_COMMON_TOKEN    stop;
<else>
  <recognizer.ASTLabelType>       start;
  <recognizer.ASTLabelType>       stop;
<endif>
  <@ruleReturnMembers()>
  <ruleDescriptor.returnScope.attributes:{<it.decl>;}; separator="\n">
}
  <ruleDescriptor.grammar.recognizerName>_<ruleDescriptor:returnStructName()>;<\n><\n>
<endif>
<endif>
>>

parameterScope(scope)
 ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name>=<expr>;"

/** Note that the scopeAttributeRef does not have access to the
* grammar name directly
*/
scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get( ctx->SCOPE_STACK(<scope>), ctx-
>SCOPE_STACK(<scope>)->size(ctx->SCOPE_STACK(<scope>)) - <negIndex> - 1) ))-><attr.name>
<else>
<if(index)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get(ctx->SCOPE_STACK(<scope>),
(ANTLR3_UINT32)<index> ) ))-><attr.name>
<else>
 (SCOPE_TOP(<scope>))-><attr.name>
<endif>
<endif>
>>
```

```
scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get( ctx->SCOPE_STACK(<scope>), ctx-
>SCOPE_STACK(<scope>)->size(ctx->SCOPE_STACK(<scope>)) - <negIndex> - 1) ))-><attr.name>
 = <expr>;
<else>
<if(index)>
 ((SCOPE_TYPE(<scope>))(ctx->SCOPE_STACK(<scope>)->get(ctx->SCOPE_STACK(<scope>),
(ANTLR3_UINT32)<index> ) ))-><attr.name> = <expr>;
<else>
 (SCOPE_TOP(<scope>))-><attr.name>=<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "ctx->SCOPE_STACK(<scope>)"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<scope>.<attr.name>
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>=<expr>;
<else>
<attr.name>=<expr>;
<endif>
>>

/**
```

```
 How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach
//
tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>->getText(<scope>))"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>->getType(<scope>))"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>->getLine(<scope>))"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>->getCharPositionInLine(<scope>))"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>->getChannel(<scope>))"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>->getTokenIndex(<scope>))"
tokenLabelPropertyRef_tree(scope,attr) ::= "(<scope>->tree)"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>->getText(<scope>)->toInt32(<scope>->getText(<scope>)))"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>.start)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>.stop)"
ruleLabelPropertyRef_tree(scope,attr)
 ::= "(<scope>.tree)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(STRSTREAM->toStringSS(STRSTREAM, <scope>.start, <scope>.start))
<else>
(STRSTREAM->toStringTT(STRSTREAM, <scope>.start, <scope>.stop))
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "<scope>.st"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "(<scope>->getType(<scope>))"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "(<scope>->getLine(<scope>))"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "(<scope>->getCharPositionInLine(<scope>))"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "(<scope>->getChannel(<scope>))"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "(<scope>->getTokenIndex(<scope>))"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "(<scope>->getText(<scope>))"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr)
 ::= "retval.stop"
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= <<
```

```
<if(TREE_PARSER)>
INPUT->toStringSS(INPUT, ADAPTOR->getTokenStartIndex(ADAPTOR, retval.start), ADAPTOR-
>getTokenStopIndex(ADAPTOR, retval.start))
<else>
STRSTREAM->toStringTT(STRSTREAM, retval.start, LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "LEXER->getText(LEXER)"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "LEXSTATE->tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "LEXSTATE->tokenStartCharPositionInLine"
lexerRulePropertyRef_channel(scope,attr) ::= "LEXSTATE->channel"
lexerRulePropertyRef_start(scope,attr) ::= "LEXSTATE->tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(LEXER->getCharIndex(LEXER)-1)"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_int(scope,attr)
 ::= "LEXER->getText(LEXER)->toInt32(LEXER->getText(LEXER))"


// setting $st and $tree is allowed in local rule. everything else is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree=<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st=<expr>;"


/** How to execute an action (when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if ( <actions.(actionScope).synpredgate> )
{
  <action>
}
<else>
if ( BACKTRACKING == 0 )
{
  <action>
}
<endif>
<else>
{
  <action>
}
<endif>
>>

// M I S C (properties, etc...)
```

```
bitsetDeclare(name, words64) ::= <<

/** Bitset defining follow set for error recovery in rule state: <name>  */
static ANTLR3_BITWORD <name>_bits[] = { <words64:{ANTLR3_UINT64_LIT(<it>)}; separator=", "> };
static  ANTLR3_BITSET_LIST <name> = { <name>_bits, <length(words64)> };
>>

bitset(name, words64) ::= <<
antlr3BitsetSetAPI(&<name>);<\n>
>>

codeFileExtension() ::=
 ".c"

true() ::= "ANTLR3_TRUE"
false() ::= "ANTLR3_FALSE"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/C/C.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/DelphiTarget.java
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
```

```
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of combinations.
*/
group ASTParser;

@rule.setErrorReturnValue()
 ::= <<
// Conversion of the second argument necessary, but harmless
retval.Tree = (<ASTLabelType>)adaptor.ErrorNode(input, (IToken) retval.Start, input.LT(-1), re);
<! System.Console.WriteLine("<ruleName> returns " + ((CommonTree)retval.Tree).ToStringTree()); !>
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if ( state.backtracking == 0 )
{
<endif>
 <label>_tree = <createNodeFromToken(...)>;
 adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>
}
<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"

/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if ( state.backtracking == 0 )
{
<endif>
 <label>_tree = <createNodeFromToken(...)>;
 root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>
}
<endif>
>>

/**
```

```
 ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,hetero,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root
 code or ruleroot code
// plus list label plus ...  The combinations might require complicated
// rather than just added on code.  Investigate that refactoring when
// I have more time.

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
<super.matchSet(..., postmatchCode={<if(backtracking)>if ( state.backtracking == 0 )
<endif>adaptor.AddChild(root_0, <createNodeFromToken(...)>);})>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<if(label)>
```

```
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<super.matchSet(..., postmatchCode={<if(backtracking)>if ( state.backtracking == 0 ) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<createNodeFromToken(...)>, root_0);})>
>>

// RULE REF
 AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking == 0 ) <endif>adaptor.AddChild(root_0, <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking == 0 ) <endif>root_0 =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_0);
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

//
 WILDCARD AST

wildcard(label,elementIndex) ::= <<
<super.wildcard(...)>
```

```
<if(backtracking)>
if ( state.backtracking == 0 )
{
<endif>
 <label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
 adaptor.AddChild(root_0, <label>_tree);
<if(backtracking)>
}
<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>
if ( state.backtracking == 0 )
{
<endif>
 <label>_tree = (<ASTLabelType>)adaptor.Create(<label>);
 root_0 = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_0);
<if(backtracking)>
}
<endif>
>>

createNodeFromToken(label,hetero) ::= <<
<if(hetero)>
new <hetero>(<label>) <! new MethodNode(IDLabel) !>
<else>
(<ASTLabelType>)adaptor.Create(<label>)
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
<endif>
 retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(!TREE_PARSER)>
 adaptor.SetTokenBoundaries(retval.Tree,
 (IToken) retval.Start, (IToken) retval.Stop);
<endif>
<if(backtracking)>
}
<endif>
```

>>

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/ASTParser.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/ASTParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD license"]
Copyright (c) 2006 Martin Traverso
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
group Ruby implements ANTLRCore;

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
         docComment, recognizer, name, tokens, tokenNames, rules,
         cyclicDFAs, bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
   scopes, superClass, literals)
 ::=
<<
# <name> (<fileName>)
# Generated by ANTLR <ANTLRVersion> on <generatedTimestamp>

<docComment>
<recognizer>
>>

/**
* Inherits parameters from outputFile(...)
*
* labelType is not used for Ruby (no explicit type declarations)
*/
lexer(grammar, name, tokens, scopes, rules, numRules, labelType, filterMode,
     superClass) ::=
<<
class <name>
   require 'stringio'

 <tokens:{<it.name>=<it.type>}; separator="\n">

   def initialize(input)
       input = StringIO.new(input) if input.respond_to?(:to_str)
       @input = CharStream.new(input)
       @backtracking = 0
       @failed = false

       <actions.lexer.init>
   end

   def next_token
    # TODO: catch exceptions
  @token = nil
  @channel = nil
  @text = nil

  @start = @input.index
  @line = @input.line
  @pos = @input.column

  @type = nil
  @type_int = nil

  return :EOF if <LA(1)> == :EOF

  match_Tokens()
  if @token == nil
   @text ||= @input.substring(@start, @input.index - 1)
   @token = Token.new(@type,
 @type_int, @line, @pos, @text, @channel)
  end

  <if(trace)>
   puts @token.inspect
  <endif>
       return @token
   end

   class Token
       attr_reader :token_type
       attr_reader :int_type
       attr_reader :line
       attr_reader :pos
       attr_reader :text
       attr_reader :channel

       def initialize(token_type, int_type, line, pos, text, channel = nil)
           @token_type = token_type
           @int_type = int_type
           @line = line
           @pos = pos
           @text = text
           @channel = channel
       end

  alias :to_i :int_type
   end

   <actions.lexer.members>

   private

   class CharStream
       attr_reader :line
       attr_reader :column
       attr_reader :index

       def initialize(input)
           @buffer = ""
           @input = input
           @line = 1
           @column = 0
```

```
     @index = 0;
   end

   # returns a Fixnum between 0 and 0xFFFF or :EOF
   def look_ahead(pos)
      offset = @index + pos - 1
     if @buffer.length \< offset + 1
       char = @input.read(offset + 1 - @buffer.length)
       @buffer \<\< char if not char.nil?
     end

     if offset \< @buffer.length
       @buffer[offset]
     else
       :EOF
     end
   end

   def mark
     @state = { :index => @index, :line => @line, :column => @column }
     return 0
   end

   def rewind(marker)
     @index = @state[:index]
     @line = @state[:line]
     @column = @state[:column]
   end

   def consume
    look_ahead(1) # force a read from the input if necessary
    @column = @column + 1
    if @buffer[@index] == ?\n
       @line = @line + 1
       @column = 0
    end
    @index = @index + 1
   end

   def substring(start, stop)
     @buffer.slice(start, stop - start
+ 1)
   end
 end


 def match(value = nil)
   @failed = false
```

```
        case
           when value.nil?
              @input.consume()
           when value.respond_to?(:to_str)
              catch(:done) do
                 value.each_byte do |c|
                    @failed ||= !(<isolatedLookaheadTest(atom="c", k=1)>)
                    @input.consume() if !@failed
                    throw :done if @failed
                 end
              end
           else
              @failed = !(<isolatedLookaheadTest(atom="value", k=1)>)
              @input.consume() if !@failed
        end

    if @failed && @backtracking \<= 0
     raise "Expected #{value.respond_to?(:chr) ? value.chr : value}"
    end
     end

     def match_range(from, to)
        char = <LA(1)>

        if char != :EOF && (char \>= from || char \<= to)
     @failed = false
     match()
        elsif @backtracking > 0
           @failed = true
        else
           raise "Expected
  [#{from.chr}..#{to.chr}]"
        end
     end

     <rules; separator="\n\n">

  <synpreds: synpred(); separator="\n\n">

  <dfaClass()>
     <cyclicDFAs: cyclicDFA()>
end
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
      bitsets, ASTLabelType, superClass,
      labelType, members) ::=
<<
```

```
require '<grammar.name>Lexer'

class <name>
 attr_reader :lexer

  TOKENS = [
    <tokenNames: {[<it>, <i>]}; separator=",\n">
  ].inject({}) { |hash, pair|
    name = pair[0]
    index = pair[1] + 3 # hardcoded for now... no way to get this value from ANTLR

    if name[0] == ?'
      hash[:"T#{index}"] = index
    else
      hash[:"#{name}"] = index
    end

    hash
  }

  TOKENS[:EOF] = -1

  def initialize(input)
    if input.respond_to?(:to_str) || input.respond_to?(:read)
      input = <grammar.name>Lexer.new(input)
    end

  @lexer = input
    @input = TokenStream.new(input)
    @backtracking
 = 0
    @failed = false

    <actions.parser.init>

    <if(trace)>
    @indent = 0
    <endif>
  end

  <rules; separator="\n\n">

  <actions.parser.members>

  private

  class TokenStream
    attr_reader :index
    def initialize(input)
      @buffer = []
      @input = input
      @channel = nil

      @index = 0;
    end

    # returns a Token
    def look_ahead(pos)
      offset = @index + pos - 1

      while @buffer[-1] != :EOF && @buffer.length \< offset + 1
        token = @input.next_token
        if token == :EOF || token.channel == @channel
          @buffer \<\< token
        end
      end

      offset = -1 if offset >= @buffer.length
      if offset \< @buffer.length
        @buffer[offset]
      end
    end

    def mark
      @state = { :index => @index }
      return 0
    end

    def
rewind(marker)
      @index = @state[:index]
    end

    def consume
     look_ahead(1) # force a read from the input if necessary
     @index = @index + 1
    end
  end

  def match(token = nil)
    if token.nil? || <LA(1)> == token
      @input.consume
      @failed = false
      return
    elsif @backtracking > 0
```

```
      @failed = true
    else
      raise "Expected #{token}"
    end
  end

  def look_ahead(k)
    token = @input.look_ahead(k)
    if token != :EOF
      token = token.token_type
    end

    token
  end

  <synpreds: synpred(); separator="\n\n">

 <dfaClass()>
  <cyclicDFAs: cyclicDFA()>
end
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
     numRules, bitsets, labelType, ASTLabelType,
     superClass, members) ::=
<<
 raise
 "treeParser not implemented"
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# <description>
def <ruleName>_fragment
  <block>
end
>>
```

```
/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::=
<<
# <description>
def <ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
 <ruleDescriptor.returnScope.attributes:
  { _retval_<it.name> = nil }; separator = "\n"
 >
 <ruleLabelDefs()>

    <if(trace)>
       puts " " * @indent + "+<ruleName>"
       @indent += 1
   <endif>

 <ruleDescriptor.actions.init>

   <block>

   <if(trace)>
       @indent -= 1
       puts " " * @indent + "-<ruleName>"
   <endif>



   <if(!ruleDescriptor.isSynPred)>
   <if(ruleDescriptor.hasReturnValue)>
   <if(ruleDescriptor.hasMultipleReturnValues)>
   return {<ruleDescriptor.returnScope.attributes:{ a | :<a.name> => _retval_<a.name> }; separator = ",">}
  # TODO: need "Attribute.index" for this to work: return <ruleDescriptor.returnScope.attributes:{ a |
_retval_<a.name> }; separator = ",">
   <else>
   return _retval_<ruleDescriptor.singleValueReturnName>
   <endif>
   <endif>
   <endif>
end
>>

ruleLabelDefs() ::= <<
<ruleDescriptor.tokenLabels :{_<it.label.text> = nil}; separator="\n">
<[ruleDescriptor.tokenListLabels, ruleDescriptor.ruleListLabels]
   :{list_<it.label.text> = nil}; separator="\n"
>
>>
```

```
/** How to generate a rule in the lexer; naked blocks are
 used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::=
<<
def match_<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
 <ruleDescriptor.actions.init>

 <lexerRuleLabelDefs()>
 <if(nakedBlock)>
  <block><\n>
 <else>
  @type = :<ruleName>
  @type_int = <ruleName>
  <block>
 <endif>
end
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels, ruleDescriptor.ruleLabels]
   :{<it.label.text> = nil}; separator="\n"
>
<ruleDescriptor.charLabels:{<it.label.text> = nil}; separator="\n">
<[ruleDescriptor.tokenListLabels, ruleDescriptor.ruleListLabels]
   :{list_<it.label.text> = nil}; separator="\n"
>
>>



/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::=
<<
<lexerRule(...)>
>>

filteringNextToken() ::=
<<
   raise "filteringNextToken not implemented"
>>

filteringActionGate() ::=
<<
```

```
   raise "filteringActionGate
 not implemented"
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
<switchBlock(...)>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
<switchBlock(...)>
>>



/**
*  decision, decisionNumber don't seem to be relevant in this template
*  alts actually has a single element
*/
ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::=
<<
<plainBlock(...)>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::=
<<
<plainBlock(...)>
>>

/** A (..)+ block with 0 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n)
 ::=
<<
# <description>
matchedOnce<decisionNumber> = false
<decls>
while true
   alt<decisionNumber> = <maxAlt>
   <decision>
   case alt<decisionNumber>
       <alts:switchCase(); separator="\n">
       else
```

```
      break
  end
  matchedOnce<decisionNumber> = true
end

if !matchedOnce<decisionNumber>
  raise "Expected at least one match: <description>"
end
>>

positiveClosureBlockSingleAlt  ::= positiveClosureBlock

/** A (..)* block with 0 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <description>
<decls>
while true
  alt<decisionNumber> = <maxAlt>
  <decision>
  case alt<decisionNumber>
    <alts:switchCase(); separator="\n">
    else
      break
  end
end
>>

closureBlockSingleAlt ::= closureBlock


/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock
 ::= block

optionalBlockSingleAlt ::= block

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::=
<<
# <description>
<elements: element(); separator="\n">
>>

// E L E M E N T S
```

```
/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::=
<<
<if(label)>
_<label> = @input.look_ahead(1)<\n>
<endif>
match(:<token>)
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::=
<<
<tokenRef(...)>
<listLabel(elem=label, ...)>
>>


listLabel(label,elem)::=
<<
list_<label> ||= []
list_<label> \<\< _<elem>
>>

/** match a character */
charRef(char,label)::=
<<
<if(label)>
_<label> = <char><\n>
<endif>
match(<char>)
>>

/** match a character range */
charRangeRef(a,b,label)::=
<<
<if(label)>
_<label> = <LA(1)><\n>
<endif>
match_range(<a>, <b>)
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode)::=
<<
<if(label)>
_<label>
 = <LA(1)><\n>
<endif>
```

```
if <s>
   match()
   <postmatchCode>
else
   raise "Expected set"
end
>>

matchSetAndListLabel(s,label,elementIndex,postmatchCode)::=
<<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label)::=
<<
<if(label)>
_<label> = <string><\n>
<endif>
match(<string>)<\n>
>>

wildcard(label, elementIndex)::=
<<
<if(label)>
_<label> = <LA(1)><\n>
<endif>
match()
>>


wildcardAndListLabel(label,elementIndex)::=
<<
<wildcard(...)>
<listLabel(elem=label,...)>
>>



/** Match . wildcard in lexer */
wildcardChar(label, elementIndex)::=
<<
<if(label)>
_<label> = <LA(1)><\n>
<endif>
match()
>>
```

```
wildcardCharListLabel(label, elementIndex)::=
<<
 raise "wildcardCharListLabel not implemented"
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.
*/
ruleRef(rule,label,elementIndex,args,scope) ::=
<<
<if(label)>
_<label> = <rule>(<args; separator=", ">)<\n>
<else>
<rule>(<args;
 separator=", ">)<\n>
<endif>
>>

/** ids+=ID */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::=
<<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>



/**

A: b=B;
B: .;

TODO: Should we use a real token type instead of :invalid? How do we get it?

*/
lexerRuleRef(rule,label,args,elementIndex,scope)::=
<<
<if(label)>
_<label>_start_<elementIndex> = @input.index
_<label>_line_<elementIndex> = @input.line
_<label>_pos_<elementIndex> = @input.column
match_<rule>(<args; separator=", ">)
_<label> = Token.new(:invalid, 0,
                   _<label>_line_<elementIndex>,
                   _<label>_pos_<elementIndex>,
                   @input.substring(_<label>_start_<elementIndex>, @input.index - 1), nil)
<else>
match_<rule>(<args; separator=", ">)
<endif>
```

```
>>


lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::=
<<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex)::=
<<
<if(label)>
_<label>
 = :EOF<\n>
<endif>
match(:EOF)
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList, enclosingTreeLevel,
    treeLevel) ::=
<<
 raise "tree not implemented"
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description)::=
<<
# <description>
if !<evalPredicate(...)>
   raise "Semantic predicate failed: #{<description>}"
end
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)::=
<<
# <description>
look_ahead<decisionNumber>_<stateNumber> = <LA(k)>
<if(LEXER)>
look_ahead<decisionNumber>_<stateNumber> = -1 if look_ahead<decisionNumber>_<stateNumber> == :EOF
<endif>

if <edges; separator="\nelsif ">
```

```
else
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt><\n>
<else>
   raise "Expected: <description>"<\n>
<endif>
end
>>

/** Same as a normal DFA state
 except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*
*  If a semPredState, don't force lookahead lookup; preds might not
*  need.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)::=
<<
# <description>
look_ahead<decisionNumber>_<stateNumber> = <LA(k)>

<if(LEXER)>
look_ahead<decisionNumber>_<stateNumber> = -1 if look_ahead<decisionNumber>_<stateNumber> == :EOF
<endif>

if <edges; separator="\nelsif ">
end
>>



/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a
*  rule anything other than 'a' predicts exiting.
*
*  If a semPredState, don't
 force lookahead lookup; preds might not
*  need.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)::=
<<
# <description>
look_ahead<decisionNumber>_<stateNumber> = <LA(k)>
<if(LEXER)>
look_ahead<decisionNumber>_<stateNumber> = -1 if look_ahead<decisionNumber>_<stateNumber> == :EOF
```

```
<endif>

if <edges; separator="\nelsif ">
<if(eotPredictsAlt)>
else
   alt<decisionNumber> = <eotPredictsAlt>
<endif>
end
>>



/** An accept state indicates a unique alternative has been predicted */
/** It is not clear that decisionNumber is available here */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates)::=
<<
<labelExpr>  <if(predicates)>&& <predicates><endif>
   <targetState>
>>

// F i x e d  D F A  (switch case)

/** A DFA state
 where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState)::=
<<
# <description>
case <LA(k)>
   <edges; separator="\n">
   else
       <if(eotPredictsAlt)>
       alt<decisionNumber> = <eotPredictsAlt><\n>
       <else>
       raise "Expected: <description>"<\n>
       <endif>
end
>>

/**
* eotPredictsAlt is not relevant here
*/
```

```
dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState)::=
<<
# <description>
case <LA(k)>
  <edges; separator="\n">
end
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState)::=
<<
# <description>
case <LA(k)>
  <edges; separator="\n">
  <if(eotPredictsAlt)><\n>
  else
     alt<decisionNumber> = <eotPredictsAlt>
  <endif>
end
>>

dfaEdgeSwitch(labels, targetState)::=
<<
<if(PARSER)>
when <labels:{:<it>}; separator=","><\n>
<else>
when
 <labels:{<it>}; separator=","><\n>
<endif>
  <targetState>
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description)::=
<<
alt<decisionNumber> = DFA<decisionNumber>.predict(self, @input)
>>

/** Generate the tables and support code needed for the DFAState object
*  argument.  Unless there is a semantic predicate (or syn pred, which
*  become sem preds), all states should be encoded in the state tables.
*  Consequently, cyclicDFAState/cyclicDFAEdge,eotDFAEdge templates are
*  not used except for special DFA states that cannot be encoded as
*  a transition table.
```

```
*/
cyclicDFA(dfa)::=
<<

DFA<dfa.decisionNumber> = DFA.new(
   [<dfa.eot; wrap="\n     ", separator=",", null="-1">],
   [<dfa.eof; wrap="\n     ", separator=",", null="-1">],
   [<dfa.min; wrap="\n     ",
 separator=",", null="0">],
   [<dfa.max; wrap="\n     ", separator=",", null="0">],
   [<dfa.accept; wrap="\n     ", separator=",", null="-1">],
   [<dfa.special; wrap="\n     ", separator=",", null="-1">],
   [
       <dfa.transition:{s | [<s; wrap="\n     ", separator=",", null="-1">]}; separator=",\n", null="">
   ])

def special_state_transition(s)
 <if(dfa.specialStateSTs)>
  case s
   <dfa.specialStateSTs:{state |
   when <i0>
    <state>}; separator="\n">
  end

  raise "Expected: <dfa.description>"
 <else>
  -1
 <endif>
end

public :special_state_transition
>>

/** A special state in a cyclic DFA; special means has a semantic predicate
*  or it's a huge set of symbols to check.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState)::=
<<
 <if(semPredState)>
 @input.rewind(0)
 <else>
 look_ahead_<decisionNumber>_<stateNumber> = <LA(1)>
 <endif>
 s = -1
 <edges>
 return s if s >= 0
>>
```

```
/** Just like a fixed DFA edge, test the lookahead and indicate
 what
*  state to jump to next if successful.  Again, this is for special
*  states.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates)::=
<<
return s = <targetStateNumber> if (<labelExpr>) <if(predicates)>&& (<predicates>)<endif><\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates)::=
<<
s = <targetStateNumber><\n>
>>

// D F A  E X P R E S S I O N S

andPredicates(left,right)::= "(<left> && <right>)"

orPredicates(operands)::=
<<
(<operands; separator=" || ">)
>>

notPredicate(pred)::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description)::= "(<pred>)"

evalSynPredicate(pred,description)::= "<pred>()"

/**
*  It's not really clear that decisionNumber and stateNumber are available here
*/
lookaheadTest(atom,k,atomAsInt)::=
<<
<if(LEXER)>
look_ahead<decisionNumber>_<stateNumber> == <atom>
<else>
look_ahead<decisionNumber>_<stateNumber> == :<atom>
<endif>
>>



/**
```

```
 Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::=
<<
<if(LEXER)>
<LA(k)> == <atom>
<else>
<LA(k)> == :<atom>
<endif>
>>

/**
*  It's not really clear that decisionNumber and stateNumber are available here
*/
lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt)::=
<<
<if(LEXER)>
(look_ahead<decisionNumber>_<stateNumber> \>= <lower> && look_ahead<decisionNumber>_<stateNumber>
\<= <upper>)
<else>
(TOKENS[look_ahead<decisionNumber>_<stateNumber>] \>= <lowerAsInt> &&
TOKENS[look_ahead<decisionNumber>_<stateNumber>] \<= <upperAsInt>)
<endif>
>>


isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::=
<<
<if(LEXER)>
(<LA(k)> \>= <lower> && <LA(k)> \<= <upper>)
<else>
(TOKENS[<LA(k)>] \>= <lowerAsInt> && TOKENS[<LA(k)>] \<= <upperAsInt>)
<endif>
>>

setTest(ranges) ::=
<<
<ranges; separator=" || ">
>>

// A T
 T R I B U T E S

parameterAttributeRef(attr)::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>"

scopeAttributeRef(scope,attr,index,negIndex)::=
<<
 raise "scopeAttributeRef not implemented"
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::=
<<
 raise "scopeSetAttributeRef not implemented"
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope)::=
<<
 raise "isolatedDynamicScopeRef not implemented"
>>

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr)::=
<<
<if(referencedRule.hasMultipleReturnValues)>
_<scope>[:<attr.name>]
<else>
_<scope>
<endif>
>>

/**
<if(referencedRule.hasMultipleReturnValues)>
<scope>[<attr.index>] # TODO: need "Attribute.index" for this to work
<else>
<scope>
<endif>
>>
**/

returnAttributeRef(ruleDescriptor,attr)::=
<<
 _retval_<attr.name>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr)
 ::=
<<
 _retval_<attr.name> = <expr>
>>
```

```
/** How to translate $tokenLabel */
tokenLabelRef(label)::= "_<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label)::= "list_<label>"


// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr)::= "_<scope>.text"
tokenLabelPropertyRef_type(scope,attr)::= "_<scope>.token_type"
tokenLabelPropertyRef_line(scope,attr)::= "_<scope>.line"
tokenLabelPropertyRef_pos(scope,attr) ::= "_<scope>.pos"
tokenLabelPropertyRef_channel(scope,attr)::= "_<scope>.channel"
tokenLabelPropertyRef_index(scope,attr)::= "_<scope>.index"


tokenLabelPropertyRef_tree(scope,attr)::= <<
 raise "tokenLabelPropertyRef_tree not implemented"
>>

ruleLabelPropertyRef_start(scope,attr)::=
<<
 raise "ruleLabelPropertyRef_start not
 implemented"
>>

ruleLabelPropertyRef_stop(scope,attr)::=
<<
 raise "ruleLabelPropertyRef_stop not implemented"
>>

ruleLabelPropertyRef_tree(scope,attr)::=
<<
 raise "ruleLabelPropertyRef_tree not implemented"
>>

ruleLabelPropertyRef_text(scope,attr)::=
<<
 raise "ruleLabelPropertyRef_text not implemented"
>>

ruleLabelPropertyRef_st(scope,attr)::=
<<
 raise "ruleLabelPropertyRef_st not implemented"
>>
```

```
/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label)::=
<<
 raise "lexerRuleLabel not implemented"
>>

lexerRuleLabelPropertyRef_type(scope,attr)::=
<<
 raise "lexerRuleLabelPropertyRef_type not implemented"
>>

lexerRuleLabelPropertyRef_line(scope,attr)::=
<<
 raise "lexerRuleLabelPropertyRef_line not implemented"
>>

lexerRuleLabelPropertyRef_pos(scope,attr)::=
<<
 raise "lexerRuleLabelPropertyRef_pos not implemented"
>>

lexerRuleLabelPropertyRef_channel(scope,attr)::=
<<
 raise "lexerRuleLabelPropertyRef_channel not implemented"
>>

lexerRuleLabelPropertyRef_index(scope,attr)::=
<<
 raise
 "lexerRuleLabelPropertyRef_index not implemented"
>>

lexerRuleLabelPropertyRef_text(scope,attr)::=
<<
 raise "lexerRuleLabelPropertyRef_text not implemented"
>>

lexerRulePropertyRef_text(scope,attr) ::= "@text"
lexerRulePropertyRef_type(scope,attr) ::= <<
 raise "lexerRulePropertyRef_type not implemented"
>>

lexerRulePropertyRef_line(scope,attr) ::= "@line"
lexerRulePropertyRef_pos(scope,attr) ::= "@pos"

lexerRulePropertyRef_index(scope,attr) ::= <<
```

```
 raise "lexerRulePropertyRef_index not implemented"
>>
lexerRulePropertyRef_channel(scope,attr) ::= "@channel"

lexerRulePropertyRef_start(scope,attr) ::= "@start"
lexerRulePropertyRef_stop(scope,attr) ::= <<
 raise "lexerRulePropertyRef_stop not implemented"
>>



ruleSetPropertyRef_tree(scope,attr,expr) ::= <<
 raise "ruleSetPropertyRef_tree not implemented"
>>
ruleSetPropertyRef_st(scope,attr,expr) ::= <<
 raise "ruleSetPropertyRef_st not implemented"
>>



// Somebody may ref $template or $tree or $stop
 within a rule:
rulePropertyRef_start(scope,attr)::=
<<
 raise "rulePropertyRef_start not implemented"
>>

rulePropertyRef_stop(scope,attr)::=
<<
 raise "rulePropertyRef_stop not implemented"
>>

rulePropertyRef_tree(scope,attr)::=
<<
 raise "rulePropertyRef_tree not implemented"
>>

rulePropertyRef_text(scope,attr)::=
<<
 raise "rulePropertyRef_text not implemented"
>>



rulePropertyRef_st(scope,attr)::=
<<
 raise "rulePropertyRef_st not implemented"
>>

/** How to execute an action */
/** TODO: add syntactic predicate & bactracking gates **/
```

```
execAction(action)::=
<<
<action>
>>

// M I S C (properties, etc...)

codeFileExtension()::=".rb"

true()::= "true"
false()::= "false"

noRewrite(rewriteBlockLevel, treeLevel) ::= ""

///////////// --------------------------- private templates --------------------------------



bitset()::=
<<
 raise "bitset not implemented"
>>



element() ::= "<it.el>"

plainBlock(decls, alts, description) ::=
<<
<decls>
<alts>
>>

switchBlock(description, decisionNumber,
 maxAlt, alts, decls, decision) ::=
<<
# <description>
alt<decisionNumber> = <maxAlt>
<decls>
<decision>
case alt<decisionNumber>
   <alts:switchCase(); separator="\n">
end
>>

switchCase() ::=
<<
when <i>
   <it>
>>
```

```
LA(k) ::=
<<
<if(LEXER)>
@input.look_ahead(<k>)
<else>
look_ahead(<k>)
<endif>
>>



synpred(name) ::= <<
def <name>
   start = @input.mark()
   @backtracking += 1
   <name>_fragment()
   @backtracking -= 1

   success = !@failed
   @input.rewind(start)
   @failed = false

   return success
end
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>



dfaClass() ::= <<
<if(cyclicDFAs)>
   class DFA
       def initialize(eot, eof, min, max, accept, special, transition)
           @eot = eot
           @eof = eof
           @min = min
           @max = max
           @accept = accept
           @special = special
           @transition = transition
       end

       def predict(parser, input)
           mark = input.mark()
            s = 0 # we always start at s0
           begin
```

```
    loop do
      special_state = @special[s]
      if special_state >= 0
        s = parser.special_state_transition(special_state)
        input.consume()
        next
      end

      if @accept[s] >= 1
        return @accept[s]
      end

      # look for a normal char transition
      c = input.look_ahead(1).to_i
      if c != :EOF && c >= @min[s] && c \<= @max[s]
        next_state = @transition[s][c - @min[s]] # move to next state
        if next_state \< 0
          # was in range but not a normal transition
          # must check EOT, which is like the else clause.
          # eot[s]>=0 indicates that an EOT edge goes to another

       # state.
          if @eot[s] >= 0  # EOT Transition to accept state?
            s = @eot[s]
            input.consume()
            next
          end
          raise "No viable alt"
        end
        s = next_state
        input.consume()
        next
      end
      if @eot[s] >= 0   # EOT Transition?
        s = @eot[s]
        input.consume()
        next
      end
      if c == :EOF && @eof[s] >= 0   # EOF Transition to accept state?
        return @accept[@eof[s]]
      end

      # not in range and not EOF/EOT, must be invalid symbol
      raise "No viable alt"
    end
  ensure
    input.rewind(mark)
```

```
            end
        end
    end
    <endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Ruby/Ruby.stg
No license file was found, but licenses were detected in source scan.

```
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
```

```
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;<\n>
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor
 property that knows how to build trees */
parserMembers() ::= <<
protected ITreeAdaptor adaptor = new CommonTreeAdaptor();<\n>
public ITreeAdaptor TreeAdaptor
{
 get
 {
  return adaptor;
 }
 set
 {
  this.adaptor = value;
  <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
}
>>

@returnScope.ruleReturnMembers() ::= <<
public <ASTLabelType> tree;
public override object Tree { get { return tree; } }
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
 :{<ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
 :{RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token
 <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
 :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule <it>");}; separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)adaptor.Nil();<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the
 tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.Add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>
```

```
/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.Add(<label>.Tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

/**
 ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

{
// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels:
 <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.Nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
input.ReplaceChildren(adaptor.GetParent(retval.start),
                      adaptor.GetChildIndex(retval.start),
                      adaptor.GetChildIndex(_last),
                      retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;<\n>
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}<\n>
<endif>
}

>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
    :{RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
   separator="\n"
>
<referencedWildcardLabels
 :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
 separator="\n"
```

```
>
<referencedWildcardListLabels
 :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",list_<it>);};
 separator="\n"
>
<referencedRuleLabels
   :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.tree:null);};
   separator="\n"
>
<referencedRuleListLabels
   :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it>",list_<it>);};
   separator="\n"
>
>>

/** Generate code for an
 optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElementsDeep:{el | stream_<el>.Reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>
```

```
rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description)
 ::=
<<
if ( !(<referencedElements:{el | stream_<el>.HasNext}; separator="||">) )
{
 throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.HasNext}; separator="||"> )
{
 <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
 <a.alt>
}<\n>
<else>
{
 <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
```

```
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
adaptor.AddChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/**
 Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, <createImaginaryNode(tokenType=token,
 ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>
```

```
/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<rule>.NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<rule>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.AddChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action)
 ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>
```

```
createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<!
 new MethodNode(IDLabel, args) !>
new <hetero>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.NextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.Create(<token>, <args; separator=", ">)
<else>
stream_<token>.NextNode()
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/AST.stg

No license file was found, but licenses were detected in source scan.

```
*/
/** Template overrides to add debugging to normal Objective-C output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
group Dbg;

@headerFile.imports() ::= <<
<@super.imports()>
#import \<ANTLR/ANTLRDebug.h>
>>

@parserHeaderFile.ivars() ::= <<
int ruleLevel;
NSArray *ruleNames;
>>

@parserHeaderFile.methodsdecl() ::= <<
-(BOOL) evalPredicate:(NSString *)predicate matched:(BOOL)result;<\n>
>>

@genericParser.init() ::= <<
ruleNames = [[NSArray alloc] initWithObjects:<rules:{rST
 | @"<rST.ruleName>"}; separator=", ", wrap="\n ">, nil];<\n>
>>

@genericParser.dealloc() ::= <<
[ruleNames release];<\n>
>>

@genericParser.methods() ::= <<
-(BOOL) evalPredicate:(NSString *)predicate matched:(BOOL)result
{
 [debugListener semanticPredicate:predicate matched:result];
 return result;
}<\n>
>>

/* bug: can't use @super.superClassName()> */
@parserHeaderFile.superClassName() ::= "ANTLRDebug<if(TREE_PARSER)>Tree<endif>Parser"

@rule.preamble() ::= <<
```

```
@try { [debugListener enterRule:@"<ruleName>"];
if ( ruleLevel==0 ) [debugListener commence];
ruleLevel++;
[debugListener locationLine:<ruleDescriptor.tree.line> column:<ruleDescriptor.tree.column>];<\n>
>>

@rule.postamble() ::= <<
[debugListener locationLine:<ruleDescriptor.EORNode.line> column:<ruleDescriptor.EORNode.column>];<\n>
}
@finally {
  [debugListener exitRule:@"<ruleName>"];
  ruleLevel--;
  if ( ruleLevel==0 ) [debugListener terminate];
}<\n>
>>

/* these are handled in the runtime for now.
* stinks,
 but that's the easiest way to avoid having to generate two
* methods for each synpred

@synpred.start() ::= "[debugListener beginBacktrack:backtracking];"

@synpred.stop() ::= "[debugListener endBacktrack:backtracking wasSuccessful:success];"

*/

// Common debug event triggers used by region overrides below

enterSubRule() ::=
  "@try { [debugListener enterSubRule:<decisionNumber>];<\n>"

exitSubRule() ::=
  "} @finally { [debugListener exitSubRule:<decisionNumber>]; }<\n>"

enterDecision() ::=
  "@try { [debugListener enterDecision:<decisionNumber>];<\n>"

exitDecision() ::=
  "} @finally { [debugListener exitDecision:<decisionNumber>]; }<\n>"

enterAlt(n) ::= "[debugListener enterAlt:<n>];<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"
```

```
@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision() ::= "<enterDecision()>"

@ruleBlock.postdecision()
 ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
   "[debugListener recognitionException:eee];<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
   "[debugListener locationLine:<it.line> column:<it.pos>];"

@matchSet.mismatchedSetException() ::=
   "[debugListener recognitionException:mse];"

@dfaState.noViableAltException() ::= "[debugListener recognitionException:nvae];"

@dfaStateSwitch.noViableAltException()
 ::= "[debugListener recognitionException:nvae];"

dfaDecision(decisionNumber,description) ::= <<
@try {
   // isCyclicDecision is only necessary for the Profiler. Which I didn't do, yet.
   // isCyclicDecision = YES;
   <super.dfaDecision(...)>
}
@catch (ANTLRNoViableAltException *nvae) {
   [debugListener recognitionException:nvae];
   @throw nvae;
}
>>

@cyclicDFA.errorMethod() ::= <<
-(void) error:(ANTLRNoViableAltException *)nvae
{
   [[recognizer debugListener] recognitionException:nvae];
}
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
[self evalPredicate:@"<description>" result:<pred>];
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ObjC/Dbg.stg

No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add a variable
 to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _first_0 = null;
<ASTLabelType> _last = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.Tree = (<ASTLabelType>)_first_0;
if ( adaptor.GetParent(retval.Tree)!=null && adaptor.IsNil( adaptor.GetParent(retval.Tree) ) )
   retval.Tree = (<ASTLabelType>)adaptor.GetParent(retval.Tree);
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)input.LT(1);
```

```
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel>
 = (<ASTLabelType>)adaptor.GetNilNode();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = (<ASTLabelType>) <root.el.label>.Tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1) == Token.DOWN )
{
   Match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef)
 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
>>

/** ID auto construct */
```

```
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if ( state.backtracking == 0 )
{
<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
 <label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
 adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>
}
<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>
if
 ( state.backtracking == 0 )
{
<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
 <label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
 root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>
}
<endif>
```

```
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.DupTree(<label>);
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate>
 )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>


// SET AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
```

```
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree
 = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<treeLevel>, <label>.Tree);
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = (<ASTLabelType>) <label>.Tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
```

```
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking
 == 0 ) <endif>root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_<treeLevel>);
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/**
 Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.NextNode())
<else>
stream_<token>.NextNode()
<endif>
>>
```

```
ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
<endif>
 retval.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(backtracking)>
}
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/ASTTreeParser.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

```
*/
/** Template subgroup to add template rewrite output
*  If debugging, then you'll also get STDbg.stg loaded.
*/
group ST;

@outputFile.imports() ::= <<
<@super.imports()>
using Antlr.StringTemplate;
using Antlr.StringTemplate.Language;
<if(!backtracking)>
using Hashtable = System.Collections.Hashtable;
<endif>

>>

/** Add this to each rule's return value struct */
@returnScope.ruleReturnMembers() ::= <<
private
 StringTemplate st;
public StringTemplate ST    { get { return st; } set { st = value; } }
public override object Template   { get { return st; } }
public override string ToString()   { return (st == null) ? null : st.ToString(); }
>>

@genericParser.members() ::= <<
<@super.members()>
protected StringTemplateGroup templateLib =
 new StringTemplateGroup("<name>Templates", typeof(AngleBracketTemplateLexer));

public StringTemplateGroup TemplateLib
{
 get { return this.templateLib; }
 set { this.templateLib = value; }
}

/// \<summary> Allows convenient multi-value initialization:
///  "new STAttrMap().Add(...).Add(...)"
/// \</summary>
protected class STAttrMap : Hashtable
{
 public STAttrMap Add(string attrName, object value)
 {
   base.Add(attrName, value);
   return this;
 }
 public STAttrMap Add(string attrName, int value)
 {
   base.Add(attrName, value);
   return this;
 }
}
>>

/** x+=rule when output=template */
ruleRefAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Template",...)>
>>

rewriteTemplate(alts) ::= <<

// TEMPLATE REWRITE
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
 <alts:rewriteTemplateAlt(); separator="else ">
 <if(rewriteMode)><replaceTextInLine()><endif>
}
<else>
<alts:rewriteTemplateAlt(); separator="else ">
<if(rewriteMode)><replaceTextInLine()><endif>
<endif>
>>

replaceTextInLine() ::= <<
<if(TREE_PARSER)>
((TokenRewriteStream)input.TokenStream).Replace(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start),
 retval.ST);
<else>
((TokenRewriteStream)input).Replace(
 ((IToken)retval.Start).TokenIndex,
 input.LT(-1).TokenIndex,
 retval.ST);
<endif>
>>

rewriteTemplateAlt() ::= <<
// <it.description>
<if(it.pred)>
if (<it.pred>) {
   retval.ST = <it.alt>;
}<\n>
<else>
{
   retval.ST = <it.alt>;
}<\n>
<endif>
>>

rewriteEmptyTemplate(alts) ::= <<
null;
>>

/** Invoke a template with a set of attribute
 name/value pairs.
*  Set the value of the rule's template *after* having set
*  the attributes because the rule's template might be used as
*  an attribute to build a bigger template; you get a self-embedded
*  template.
*/
rewriteExternalTemplate(name,args) ::= <<
templateLib.GetInstanceOf("<name>"<if(args)>,
 new STAttrMap()<args:{a | .Add("<a.name>", <a.value>)}>
 <endif>)
>>

/** expr is a string expression that says what template to load */
rewriteIndirectTemplate(expr,args) ::= <<
templateLib.GetInstanceOf(<expr><if(args)>,
 new STAttrMap()<args:{a | .Add("<a.name>", <a.value>)}>
 <endif>)
>>

/** Invoke an inline template with a set of attribute name/value pairs */
rewriteInlineTemplate(args, template) ::= <<
new StringTemplate(templateLib, "<template>"<if(args)>,
 new STAttrMap()<args:{a | .Add("<a.name>", <a.value>)}>
 <endif>)
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
<action>
>>

/** An action has %st.attrName=expr; or %{st}.attrName=expr; */
actionSetAttribute(st,attrName,expr)
```

```
 ::= <<
(<st>).SetAttribute("<attrName>",<expr>);
>>

/** Translate %{stringExpr} */
actionStringConstructor(stringExpr) ::= <<
new StringTemplate(templateLib,<stringExpr>)
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/ST.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/ST.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
group CSharp implements ANTLRCore;

```
csharpTypeInitMap ::= [
 "int":"0",
 "uint":"0",
 "long":"0",
 "ulong":"0",
 "float":"0.0",
 "double":"0.0",
 "bool":"false",
 "byte":"0",
 "sbyte":"0",
 "short":"0",
 "ushort":"0",
 "char":"char.MinValue",
 default:"null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
 */
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@debugPreprocessor()>
<actions.(actionScope).header>

<@imports>
using System;
using Antlr.Runtime;
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
<endif>
using IList   = System.Collections.IList;
using ArrayList  = System.Collections.ArrayList;
using Stack   = Antlr.Runtime.Collections.StackList;

<if(backtracking)>
using IDictionary = System.Collections.IDictionary;
using Hashtable  = System.Collections.Hashtable;
<endif>
```

```
<@end>

<if(actions.(actionScope).namespace)>
namespace <actions.(actionScope).namespace>
{
<endif>

<docComment>
<recognizer>
<if(actions.(actionScope).namespace)>
}
<endif>
>>

lexer(grammar,
 name, tokens, scopes, rules, numRules, labelType="IToken",
     filterMode, superClass="Lexer") ::= <<
public class <grammar.recognizerName> : <@superClassName><superClass><@end> {
   <tokens:{public const int <it.name> = <it.type>;}; separator="\n">
   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
   <actions.lexer.members>

   // delegates
   <grammar.delegates:
        {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
   <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

   public <grammar.recognizerName>() <! needed by subclasses !>
   {
  InitializeCyclicDFAs();
   }
   public <grammar.recognizerName>(ICharStream input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>)
  : this(input, null<grammar.delegators:{g|, <g:delegateName()>}>)
 {
   }
   public <grammar.recognizerName>(ICharStream input, RecognizerSharedState state<grammar.delegators:{g|,
<g.recognizerName> <g:delegateName()>}>)
  : base(input, state) {
  InitializeCyclicDFAs(); <! Necessary in C#??? Not removed yet. !>
<if(memoize)>
<if(grammar.grammarIsRoot)>
       state.ruleMemo = new Hashtable[<numRules>+1];<\n> <! index from 1..n !>
<endif>
<endif>
```

```
       <grammar.directDelegates:
        {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
       <grammar.delegators:
        {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
       <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
   }

   override public string GrammarFileName
   {
    get { return "<fileName>";}
   }

<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>

   <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber>
 dfa<dfa.decisionNumber>;}; separator="\n">
 private void InitializeCyclicDFAs(<@debugInitializeCyclicDFAs()>)
 {
     <cyclicDFAs:{dfa | this.dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this<@debugAddition()>);}; separator="\n">
     <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>this.dfa<dfa.decisionNumber>.specialStateTransitionHandler = new
DFA.SpecialStateTransitionHandler(DFA<dfa.decisionNumber>_SpecialStateTransition);<endif>};
separator="\n">
 }

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
override public IToken NextToken()
{
   while (true)
 {
       if ( input.LA(1) == (int)CharStreamConstants.EOF
 )
  {
        return Token.EOF_TOKEN;
       }

     state.token = null;
 state.channel = Token.DEFAULT_CHANNEL;
       state.tokenStartCharIndex = input.Index();
       state.tokenStartCharPositionInLine = input.CharPositionInLine;
       state.tokenStartLine = input.Line;
     state.text = null;
       try
  {
        int m = input.Mark();
        state.backtracking = 1; <! means we won't throw slow exception !>
        state.failed = false;
        mTokens();
        state.backtracking = 0;
<!
   mTokens backtracks with synpred at backtracking==2
        and we set the synpredgate to allow actions at level 1.
!>
        if ( state.failed )
   {
           input.Rewind(m);
             input.Consume(); <! // advance one char and try again !>
        }
        else
   {
    Emit();
             return state.token;
        }
       }
       catch (RecognitionException re)
   {
        // shouldn't
 happen in backtracking mode, but...
        ReportError(re);
        Recover(re);
       }
   }
}

override public void Memoize(IIntStream input, int ruleIndex, int ruleStartIndex)
{
 if ( state.backtracking > 1 )
   base.Memoize(input, ruleIndex, ruleStartIndex);
```

```
}

override public bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
 if ( state.backtracking>1 )
  return base.AlreadyParsedRule(input, ruleIndex);
 return false;
}
>>

actionGate() ::= "(state.backtracking==0)"

filteringActionGate() ::= "(state.backtracking == 1)"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass, filterMode,
            ASTLabelType="object", labelType, members, rewriteElementType) ::= <<
public class <grammar.recognizerName> : <@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
   public static readonly string[] tokenNames = new string[]
 {
       "\<invalid>",

  "\<EOR>",
  "\<DOWN>",
  "\<UP>",
  <tokenNames; separator=", \n">
   };<\n>
<endif>

   <tokens:{public const int <it.name> = <it.type>;}; separator="\n">

   // delegates
   <grammar.delegates:
        {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
   <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
   <@members>
   <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>

   public <grammar.recognizerName>(<inputStreamType> input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>)
```

```
  : this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>) {
   }

   public <grammar.recognizerName>(<inputStreamType> input, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName>
 <g:delegateName()>}>)
  : base(input, state) {
      <parserCtorBody()>
      <grammar.directDelegates:
       {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
      <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
      <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
  }
   <@end>

   override public string[] TokenNames {
  get { return <grammar.composite.rootGrammar.recognizerName>.tokenNames; }
   }

   override public string GrammarFileName {
  get { return "<fileName>"; }
   }

   <members>

   <rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
   // Delegated rules
   <grammar.delegatedRules:{ruleDescriptor|
   public <returnType()> <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
 // throws RecognitionException
   \{
    <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>}; separator=", ">);
   \}}; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">
 private void InitializeCyclicDFAs(<@debugInitializeCyclicDFAs()>)
 {
    <cyclicDFAs:{dfa | this.dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this<@debugAddition()>);}; separator="\n">
```

```
    <cyclicDFAs:{dfa | <if(dfa.specialStateSTs)>this.dfa<dfa.decisionNumber>.specialStateTransitionHandler = new
DFA.SpecialStateTransitionHandler(DFA<dfa.decisionNumber>_SpecialStateTransition);<endif>};
separator="\n">
 }

  <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

  <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                words64=it.bits)>
}
>>

parserCtorBody()
 ::= <<
<@initializeCyclicDFAs>InitializeCyclicDFAs();<@end>
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new Hashtable[<length(grammar.allImportedRules)>+1];<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType, superClass="Parser",
labelType="IToken", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/**
 A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
```

```
// $ANTLR start "<ruleName>"
public void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) {
  <ruleLabelDefs()>
<if(trace)>
  TraceIn("<ruleName>_fragment", <ruleDescriptor.index>);
  try
  {
    <block>
  }
  finally
  {
    TraceOut("<ruleName>_fragment", <ruleDescriptor.index>);
  }
<else>
  <block>
<endif>
}
// $ANTLR end "<ruleName>"
>>

synpredDecls(name) ::= <<
SynPredPointer <name>;<\n>
>>

synpred(name) ::= <<
public bool <name>()
{
  state.backtracking++;
   <@start()>
  int start = input.Mark();
  try
  {
    <name>_fragment(); // can never throw exception
  }
  catch (RecognitionException re)
  {
    Console.Error.WriteLine("impossible: "+re);
  }
  bool success = !state.failed;
  input.Rewind(start);
  <@stop()>
  state.backtracking--;
  state.failed = false;
  return success;
}<\n>
>>

lexerSynpred(name) ::= <<
```

```
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( (state.backtracking > 0) && AlreadyParsedRule(input, <ruleDescriptor.index>) )
{
 return <ruleReturnValue()>;
}
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if ( state.backtracking > 0 ) {state.failed = true; return <ruleReturnValue()>;}<endif>
>>

/** How to
 generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
public <returnType()> <ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) // throws
RecognitionException [1]
{
   <if(trace)>TraceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   try
 {
     <ruleMemoization(name=ruleName)>
       <block>
       <ruleCleanUp()>
       <(ruleDescriptor.actions.after):execAction()>
   }
<if(exceptions)>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
  <actions.(actionScope).rulecatch>
<else>
   catch (RecognitionException re)
 {
     ReportError(re);
     Recover(input,re);
 <@setErrorReturnValue()>
  }<\n>
<endif>
<endif>
<endif>
  finally
 {
     <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
     <memoize()>
     <ruleScopeCleanUp()>
     <finally>
  }
  <@postamble()>
  return <ruleReturnValue()>;
}
// $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>)
{
  <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = new <returnType()>();
retval.Start = input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.Index();
<endif>
```

```
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.Push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.Push(new
 <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.Pop();}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.Pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{<labelType> <it.label.text> = null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{IList list_<it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll:ruleLabelDef(label=it)> <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.ruleListLabels,
 ruleDescriptor.ruleLabels]
   :{<labelType> <it.label.text> = null;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;};
 separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{IList list_<it.label.text> = null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
```

```
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.Stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking > 0 )
{
 Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex);
}
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
public void m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) // throws
RecognitionException [2]
{
  <ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
 <if(trace)>TraceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
  try
  {
<if(nakedBlock)>
      <ruleMemoization(name=ruleName)>
      <lexerRuleLabelDefs()>
      <ruleDescriptor.actions.init>
      <block><\n>
<else>
      int _type = <ruleName>;
 int _channel = DEFAULT_TOKEN_CHANNEL;
      <ruleMemoization(name=ruleName)>
      <lexerRuleLabelDefs()>
      <ruleDescriptor.actions.init>
      <block>
```

```
    <ruleCleanUp()>
    state.type = _type;
    state.channel = _channel;
    <(ruleDescriptor.actions.after):execAction()>
<endif>
  }
  finally
 {
    <if(trace)>TraceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeCleanUp()>
    <memoize()>
  }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor)
 ::= <<
override public void mTokens() // throws RecognitionException
{
  <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
{
  <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
```

```
int alt<decisionNumber> = <maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>)
{
   <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/**
 A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
int cnt<decisionNumber> = 0;
<decls>
<@preloop()>
do
{
   int alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
 {
  <alts:altSwitchCase()>
  default:
      if ( cnt<decisionNumber> >= 1 ) goto loop<decisionNumber>;
      <ruleBacktrackFailure()>
```

```
            EarlyExitException eee<decisionNumber> =
                new EarlyExitException(<decisionNumber>, input);
            <@earlyExitException()>
            throw eee<decisionNumber>;
    }

  cnt<decisionNumber>++;
} while (true);

loop<decisionNumber>:
 ; // Stops C# compiler whining that label 'loop<decisionNumber>' has no statements
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
do
{
   int alt<decisionNumber> = <maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>)
 {
  <alts:altSwitchCase()>
  default:
      goto loop<decisionNumber>;
   }
} while (true);

loop<decisionNumber>:
 ; // Stops C# compiler whining that label 'loop<decisionNumber>' has no statements
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block
```

```
optionalBlockSingleAlt
 ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> :
   <@prealt()>
   <it>
   break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
 <@declarations()>
 <elements:element()>
 <rew>
 <@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label>=(<labelType>)<endif>Match(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex
>); <checkRuleBacktrackFailure()>
>>

/**
 ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
```

```
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label> == null) list_<label> = new ArrayList();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label> = (<labelType>)input.LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
   input.Consume();
   <postmatchCode>
<if(!LEXER)>
   state.errorRecovery = false;
<endif>
   <if(backtracking)>state.failed = false;<endif>
}
else
{
   <ruleBacktrackFailure()>
    MismatchedSetException mse = new MismatchedSetException(null,input);
   <@mismatchedSetException()>
<if(LEXER)>
```

```
  Recover(mse);
  throw mse;
<else>
  throw mse;
  <! use following code to make it recover inline; remove throw mse;
  RecoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
  !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
int <label>Start = CharIndex;
Match(<string>); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start, CharIndex-1);
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label> = (<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex)
 ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
```

```
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)>
<label> = <if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
<else>
<if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">);<\n>
<endif>
state.followingStackPointer--;
<checkRuleBacktrackFailure()>
>>

/**
 ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = CharIndex;
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, CharIndex-1);
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
```

```
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex)
 ::= <<
<if(label)>
int <label>Start<elementIndex> = CharIndex;
Match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, CharIndex-1);
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1) == Token.DOWN )
{
   Match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description)
 ::= <<
if ( !(<evalPredicate(...)>) )
{
  <ruleBacktrackFailure()>
  throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
{
<if(eotPredictsAlt)>
  alt<decisionNumber> = <eotPredictsAlt>;
<else>
  <ruleBacktrackFailure()>
  NoViableAltException nvae_d<decisionNumber>s<stateNumber> =
      new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
  <@noViableAltException()>
  throw nvae_d<decisionNumber>s<stateNumber>;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);"
 and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
```

```
else
{
   alt<decisionNumber> = <eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::=
 "alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>) <endif>)
{
   <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
<edges; separator="\n">
 default:
<if(eotPredictsAlt)>
    alt<decisionNumber> = <eotPredictsAlt>;
    break;
<else>
     <ruleBacktrackFailure()>
     NoViableAltException nvae_d<decisionNumber>s<stateNumber> =
         new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
     <@noViableAltException()>

 throw nvae_d<decisionNumber>s<stateNumber>;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
```

```
switch ( input.LA(<k>) )
{
  <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
 default:
    alt<decisionNumber> = <eotPredictsAlt>;
    break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n">
 {
  <targetState>
  }
  break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
>>

/* Dump DFA tables.
*/
cyclicDFA(dfa) ::= <<
const string DFA<dfa.decisionNumber>_eotS
 =
  "<dfa.javaCompressedEOT; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_eofS =
  "<dfa.javaCompressedEOF; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_minS =
  "<dfa.javaCompressedMin; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_maxS =
  "<dfa.javaCompressedMax; wrap="\"+\n    \"">";
```

```
const string DFA<dfa.decisionNumber>_acceptS =
  "<dfa.javaCompressedAccept; wrap="\"+\n    \"">";
const string DFA<dfa.decisionNumber>_specialS =
  "<dfa.javaCompressedSpecial; wrap="\"+\n    \"">}>";
static readonly string[] DFA<dfa.decisionNumber>_transitionS = {
    <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
};

static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
static readonly short[] DFA<dfa.decisionNumber>_eof =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
static
 readonly char[] DFA<dfa.decisionNumber>_max =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
static readonly short[][] DFA<dfa.decisionNumber>_transition =
DFA.UnpackEncodedStringArray(DFA<dfa.decisionNumber>_transitionS);

protected class DFA<dfa.decisionNumber> : DFA
{
  <@debugMember()>
  public DFA<dfa.decisionNumber>(BaseRecognizer recognizer)
  {
    this.recognizer = recognizer;
    this.decisionNumber = <dfa.decisionNumber>;
    this.eot = DFA<dfa.decisionNumber>_eot;
    this.eof = DFA<dfa.decisionNumber>_eof;
    this.min = DFA<dfa.decisionNumber>_min;
    this.max = DFA<dfa.decisionNumber>_max;
    this.accept = DFA<dfa.decisionNumber>_accept;
     this.special = DFA<dfa.decisionNumber>_special;
    this.transition = DFA<dfa.decisionNumber>_transition;

  }
  <@dbgCtor()>

  override public string Description
  {
    get { return "<dfa.description>"; }
  }

  <@errorMethod()>
```

```
}<\n>
<if(dfa.specialStateSTs)>

protected internal int DFA<dfa.decisionNumber>_SpecialStateTransition(DFA dfa, int s, IIntStream _input)
//throws NoViableAltException
{
	<if(LEXER)>
	IIntStream input = _input;
	<endif>
	<if(PARSER)>
	ITokenStream input = (ITokenStream)_input;
	<endif>
	<if(TREE_PARSER)>
	ITreeNodeStream input = (ITreeNodeStream)_input;
	<endif>
 int _s = s;
   switch ( s )
   {
   <dfa.specialStateSTs:{state |
	case <i0> : <! compressed special state numbers 0..n-1 !>
	    <state>}; separator="\n">
   }
<if(backtracking)>
   if (state.backtracking > 0) {state.failed = true; return -1;}<\n>
<endif>
   NoViableAltException
 nvae<dfa.decisionNumber> =
	new NoViableAltException(dfa.Description, <dfa.decisionNumber>, _s, input);
   dfa.Error(nvae<dfa.decisionNumber>);
   throw nvae<dfa.decisionNumber>;
}<\n>
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(1);<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
int index<decisionNumber>_<stateNumber> = input.Index();
input.Rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.Seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
```

```
if ( s >= 0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates)
 ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>) <endif>) { s = <targetStateNumber>; }<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left> && <right>)"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o | || <o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber> == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber>
 \>= <lower> && LA<decisionNumber>_<stateNumber> \<= <upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>) \>=
<lower> && input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= "<ranges; separator=\" || \">"
```

```
// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected class <scope.name>_scope
{
   <scope.attributes:{protected internal <it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected class <scope.name>_scope
{
   <scope.attributes:{protected internal <it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/**
 Generate the C# type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>.<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
```

```
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<csharpTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop
 tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public class <ruleDescriptor:returnStructName()> :
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope
{
   <scope.attributes:{public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
};
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack[<scope>_stack.Count-<negIndex>-1]).<attr.name>
```

```
<else>
<if(index)>
((<scope>_scope)<scope>_stack[<index>]).<attr.name>
<else>
((<scope>_scope)<scope>_stack.Peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack[<scope>_stack.Count-<negIndex>-1]).<attr.name>
 = <expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack[<index>]).<attr.name> = <expr>;
<else>
((<scope>_scope)<scope>_stack.Peek()).<attr.name> = <expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
((<scope> != null) ? <scope>.<attr.name> : <initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> = <expr>;
```

```
<else>
<attr.name>
 = <expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "((<scope> != null) ? <scope>.Text : null)"
tokenLabelPropertyRef_type(scope,attr) ::= "((<scope> != null) ? <scope>.Type : 0)"
tokenLabelPropertyRef_line(scope,attr) ::= "((<scope> != null) ? <scope>.Line : 0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "((<scope> != null) ? <scope>.CharPositionInLine : 0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "((<scope> != null) ? <scope>.Channel : 0)"
tokenLabelPropertyRef_index(scope,attr) ::= "((<scope> != null) ? <scope>.TokenIndex : 0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int.Parse(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "((<scope> != null) ? ((<labelType>)<scope>.Start) : null)"
ruleLabelPropertyRef_stop(scope,attr)
 ::= "((<scope> != null) ? ((<labelType>)<scope>.Stop) : null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "((<scope> != null) ? ((<ASTLabelType>)<scope>.Tree) : null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
((<scope> != null) ? input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(<scope>.Start),
 input.TreeAdaptor.GetTokenStopIndex(<scope>.Start)) : null)
<else>
((<scope> != null) ? input.ToString((IToken)(<scope>.Start),(IToken)(<scope>.Stop)) : null)
<endif>
>>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> != null) ? <scope>.ST : null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> != null) ? <scope>.Type : 0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> != null) ? <scope>.Line : 0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> != null) ? <scope>.CharPositionInLine : -1)"
lexerRuleLabelPropertyRef_channel(scope,attr)
 ::= "((<scope> != null) ? <scope>.Channel : 0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "((<scope> != null) ? <scope>.TokenIndex : 0)"
```

```
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> != null) ? <scope>.Text : null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?int.Parse(<scope>.Text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.Start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType>)retval.Stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.Tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(retval.Start),
 input.TreeAdaptor.GetTokenStopIndex(retval.Start) )
<else>
input.ToString((IToken)retval.Start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.ST"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr)
 ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(CharIndex-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.Tree = <expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.ST = <expr>;"


/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
 <action>
}
<else>
<action>
<endif>
>>
```

```
/** How to always execute an action even when backtracking */
execForcedAction(action)
 ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static readonly BitSet <name> = new BitSet(new ulong[]{<words64:{<it>UL};separator=",">});<\n>
>>

codeFileExtension() ::= ".cs"

true() ::= "true"
false() ::= "false"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/CSharp.stg
No license file was found, but licenses were detected in source scan.

```
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
import org.antlr.runtime.tree.*;<\n>
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected var adaptor:TreeAdaptor = new CommonTreeAdaptor();<\n>
override public function set treeAdaptor(adaptor:TreeAdaptor):void {
   this.adaptor = adaptor;
    <grammar.directDelegates:{g|<g:delegateName()>.treeAdaptor = this.adaptor;}>
}
override public function get treeAdaptor():TreeAdaptor {
   return adaptor;
}
>>

@returnScope.ruleReturnMembers() ::= <<
<ASTLabelType> tree;
public function get tree():Object { return tree; }
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var root_0:<ASTLabelType> = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
 ruleDescriptor.wildcardTreeListLabels]:{var <it.label.text>_tree:<ASTLabelType>=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{var <it.label.text>_tree:<ASTLabelType>=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{var stream_<it>:RewriteRuleSubtreeStream=new
 RewriteRuleSubtreeStream(adaptor,"rule <it>");}; separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = <ASTLabelType>(adaptor.nil());<\n>
<endif>
<endif>
<endif>
>>

// T r a c k i n g  R u l e  E l e m e n t s

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/**
 ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
```

```
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".tree",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".tree",...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.tree
```

```
 = root_0;
<rewriteCodeLabels()>
root_0 = <ASTLabelType>(adaptor.nil());
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = <ASTLabelType>(adaptor.rulePostProcessing(root_0));
input.replaceChildren(adaptor.getParent(retval.start),
                      adaptor.getChildIndex(retval.start),
                      adaptor.getChildIndex(_last),
                      retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{var stream_<it>:RewriteRule<rewriteElementType>Stream=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{var stream_<it>:RewriteRule<rewriteElementType>Stream=new
 RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
   separator="\n"
>
<referencedWildcardLabels
   :{var  stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
   separator="\n"
>
<referencedWildcardListLabels
   :{var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"wildcard
<it>",list_<it>);};
   separator="\n"
>
<referencedRuleLabels
```

```
  :{var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.tree:null);};
  separator="\n"
>
<referencedRuleListLabels
  :{var stream_<it>:RewriteRuleSubtreeStream=new RewriteRuleSubtreeStream(adaptor,"rule <it>",list_<it>);};
  separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all
 nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext}; separator="||"> ) {
  <alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext}; separator="||"> ) {
  <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext}; separator="||">) ) {
  throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.hasNext};
 separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
var root_<treeLevel>:<ASTLabelType> = <ASTLabelType>(adaptor.nil());
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
```

```
adaptor.addChild(root_<treeLevel>,
 stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>));<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>));<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<createImaginaryNode(tokenType=token,
 ...)>, root_<treeLevel>));<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>
```

```
rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>));<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<action>, root_<treeLevel>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label)
 ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>));<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel, args) !>
new <hetero>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
<ASTLabelType>(adaptor.create(<tokenType>, <args;
 separator=", "><if(!args)>"<tokenType>"<endif>))
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
```

```
<if(hetero)>
new <hetero>(stream_<token>.nextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ActionScript/AST.stg

No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*
 is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _first_0 = null;
<ASTLabelType> _last = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( adaptor.GetParent(retval.tree)!=null && adaptor.IsNil( adaptor.getParent(retval.tree) ) )
   retval.tree = (<ASTLabelType>)adaptor.GetParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType>
 _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.Nil();
<endif>
```

```
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
   Match(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   Match(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

//
 TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
```

```
<label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto
 construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.DupTree(<label>);
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
```

```
<else>
 <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
adaptor.AddChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.DupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.AddChild(root_<treeLevel>, <label>.Tree);
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.BecomeRoot(<label>.Tree, root_<treeLevel>);
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>
```

```
/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change NextToken to NextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.NextNode())
<else>
stream_<token>.NextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/GrammarSpelunker.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/Graph.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/BuildDependencyGenerator.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/Tool.java
No license file was found, but licenses were detected in source scan.

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
group ASTDbg;

/*
parserMembers() ::= <<
protected TreeAdaptor adaptor =
   new DebugTreeAdaptor(dbg,new CommonTreeAdaptor());
public void setTreeAdaptor(TreeAdaptor adaptor) {
   this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
}
public TreeAdaptor getTreeAdaptor() {
   return adaptor;
}<\n>
>>
*/

@treeParserHeaderFile.superClassName ::= "ANTLRDebugTreeParser"

@rewriteElement.pregen() ::= "[debugListener locationLine:<e.line> column:<e.pos>];"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ObjC/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions

```
group AST;

/** Add an adaptor property that knows how to build trees */
@headerFile.members() ::= <<
/* @headerFile.members() */
pANTLR3_BASE_TREE_ADAPTOR adaptor;
pANTLR3_VECTOR_FACTORY  vectors;
/* End @headerFile.members() */
>>

/** Install the tree adaptor interface pointer and anything else that
*  tree parsers and producers require.
*/
@genericParser.apifuncs() ::= <<
<if(PARSER)>
ADAPTOR =
 ANTLR3_TREE_ADAPTORNew(instream->tstream->tokenSource->strFactory);<\n>
<endif>
ctx->vectors = antlr3VectorFactoryNew(0);
>>

@genericParser.cleanup() ::= <<
ctx->vectors->close(ctx->vectors);
<if(PARSER)>
/* We created the adaptor so we must free it
*/
ADAPTOR->free(ADAPTOR);
```

```
<endif>
>>

@returnScope.ruleReturnMembers() ::= <<

<super.ASTLabelType()> tree;

>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0;<\n>
>>

ruleInitializations() ::= <<
<super.ruleInitializations()>
root_0 = NULL;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{<ASTLabelType> <it.label.text>_tree;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<ASTLabelType> <it.label.text>_tree;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
  :{pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
  :{pANTLR3_REWRITE_RULE_SUBTREE_STREAM
 stream_<it>;}; separator="\n">
>>

/* Note that we defer the actual creation of any rewrite streams we need here and just initialize
* them to NULL. This saves creating huge numbers of rewrite streams that cannot be used as only
* one alt will be taken in a rule, but we are declaring all the streams here. So we define
* a macro that conatins the create code, then use this macro later to check if the stream
* has been created yet. Checking for NULL is almost free in C.
*/
ruleLabelInitializations() ::= <<
<super.ruleLabelInitializations()>
<ruleDescriptor.tokenLabels:{<it.label.text>_tree   = NULL;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<it.label.text>_tree   = NULL;}; separator="\n">

<ruleDescriptor.allTokenRefsInAltsWithRewrites
:{stream_<it>   = NULL;
#define CREATE_stream_<it>  if (stream_<it> == NULL) {stream_<it> =
antlr3RewriteRule<rewriteElementType>StreamNewAE(ADAPTOR, RECOGNIZER, (pANTLR3_UINT8)"token
<it>"); } }; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
```

```
:{stream_<it>
   = NULL;
#define CREATE_stream_<it>  if (stream_<it> == NULL) {stream_<it> =
antlr3RewriteRuleSubtreeStreamNewAE(ADAPTOR, RECOGNIZER, (pANTLR3_UINT8)"rule <it>"); }};
separator="\n">

<if(ruleDescriptor.hasMultipleReturnValues)>
retval.tree  = NULL;
<endif>
>>

/**  a rule label including default value */
ruleLabelInitVal(label) ::= <<
<super.ruleLabelInitVal(...)>
<label.label.text>.tree = <initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<endif>
<endif>
>>

@alt.initializations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));<\n>
<endif>
<endif>
<endif>
>>



//
 Tracking Rule Elements
//
/** ID but track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<token>;
stream_<token>->add(stream_<token>, <label>, NULL);<if(backtracking)> }<endif><\n>
```

```
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>CREATE_stream_<token>; stream_<token>-
>add(stream_<token>, <label>, NULL);<if(backtracking)> }<endif><\n>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>



/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<rule.name>;
stream_<rule.name>->add(stream_<rule.name>, <label>.tree, NULL);<if(backtracking)> }<endif>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabelTrack(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) { <endif>CREATE_stream_<rule.name>;
stream_<rule.name>->add(stream_<rule.name>, <label>.tree, NULL);<if(backtracking)> }<endif>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabelAST(...)>
>>
```

```
// RULE REF AST



/** Match
 ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>


/* How to accumulate lists when we are doing rewrite tracking...
*/
listLabelTrack(label) ::= <<
/* listLabelTrack(label)
*/
if (list_<label> == NULL)
{
   list_<label>=ctx->vectors->newVector(ctx->vectors);
}
list_<label>->add(list_<label>, <label>.tree, NULL);
>>

/* How to accumulate lists of rule outputs (only allowed with AST
* option but if the user is going to walk the tree, they will want
* all their custom elements from rule returns.
*
* Normally, we use inline structures (which the compiler lays down
* code to copy from heap allocations. However, here we want to accumulate copies
* of the returned structures because we are adding them to a list. This only makes sense if the
* grammar is not rewriting the tree as a tree rewrite only preserves the tree, not the object/structure
* returned from
 the rule. The rewrite will extract the tree pointer. However, if we are not going to
* do a tree re-write, then the user may wish to iterate the structures returned by the rule in
* action code and will expect the user defined returns[] elements to be available when they do this.
* Hence we cannot just preserve the tree that was returned. So, we must copy the local structure and provide
* a function that can free the allocated space. We cannot know how to free user allocated elements and
* presume that the user will know to do this using their own factories for the structures they allocate.
*/
listLabelAST(label) ::= <<
if (list_<label> == NULL)
{
   list_<label>=ctx->vectors->newVector(ctx->vectors);
}
{
   RETURN_TYPE_<label> * tcopy;
   tcopy = ANTLR3_MALLOC(sizeof(RETURN_TYPE_<label>)); /* Note no memory allocation checks! */
   ANTLR3_MEMCPY((void *)(tcopy), (const void *)&<label>, sizeof(RETURN_TYPE_<label>));
   list_<label>->add(list_<label>, tcopy, freeScope);
  /* Add whatever the return type is */<\n>
}
>>

// Rewrite

rewriteCode(
 alts,
 description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel,
 enclosingTreeLevel,
 treeLevel) ::=
<<

/* AST REWRITE
* elements          : <referencedElementsDeep; separator=", ">
* token labels      : <referencedTokenLabels; separator=", ">
* rule labels       : <referencedRuleLabels; separator=", ">
* token list labels : <referencedTokenListLabels; separator=", ">
* rule list labels  : <referencedRuleListLabels; separator=", ">
*/
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) <\n>
<endif>
{
 <rewriteCodeLabelsDecl()>
 <rewriteCodeLabelsInit()>
 root_0       = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
 <prevRuleRootRef()>.tree    = root_0;
 <alts:rewriteAlt(); separator="else
 ">
 <if(TREE_PARSER)>
 <if(rewriteMode)>
 <prevRuleRootRef()>.tree = (<ASTLabelType>)(ADAPTOR->rulePostProcessing(ADAPTOR, root_0));
 INPUT->replaceChildren(INPUT, ADAPTOR->getParent(ADAPTOR, retval.start),
                    ADAPTOR->getChildIndex(ADAPTOR, retval.start),
                    ADAPTOR->getChildIndex(ADAPTOR, _last),
                    retval.tree);
 <endif>
 <endif>
 <prevRuleRootRef()>.tree = root_0; // set result root
 <rewriteCodeLabelsFree()>

}
>>

rewriteCodeLabelsDecl() ::= <<
<referencedTokenLabels
   :{pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;};
   separator="\n"
>
<referencedTokenListLabels
   :{pANTLR3_REWRITE_RULE_<rewriteElementType>_STREAM stream_<it>;};
   separator="\n"
>
<referencedRuleLabels
   :{pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;};
   separator="\n"
>
<referencedRuleListLabels
   :{pANTLR3_REWRITE_RULE_SUBTREE_STREAM stream_<it>;};
   separator="\n"
>
>>

rewriteCodeLabelsInit() ::= <<
<referencedTokenLabels
:{stream_<it>=antlr3RewriteRule<rewriteElementType>StreamNewAEE(ADAPTOR,
  RECOGNIZER, (pANTLR3_UINT8)"token <it>", <it>);};
separator="\n"
>
<referencedTokenListLabels
:{stream_<it>=antlr3RewriteRule<rewriteElementType>StreamNewAEV(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", list_<it>); };
separator="\n"
>
<referencedRuleLabels
:{stream_<it>=antlr3RewriteRuleSubtreeStreamNewAEE(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", <it>.tree != NULL ? <it>.tree : NULL);};
separator="\n"
>
<referencedRuleListLabels
:{stream_<it>=antlr3RewriteRuleSubtreeStreamNewAEV(ADAPTOR,  RECOGNIZER,
(pANTLR3_UINT8)"token <it>", list_<it>);};
separator="\n"
```

```
>
>>
rewriteCodeLabelsFree() ::= <<
<referencedTokenLabels
:{if (stream_<it> != NULL) stream_<it>->free(stream_<it>); };
separator="\n"
>
<referencedTokenListLabels
:{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
<referencedRuleLabels
:{if (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
<referencedRuleListLabels
:{if
 (stream_<it> != NULL) stream_<it>->free(stream_<it>);};
separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,
 rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,  // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
{
 if ( <referencedElementsDeep:{el | (stream_<el> != NULL && stream_<el>->hasNext(stream_<el>)) };
separator="|| "> )
 {
  <alt>
 }
 <referencedElementsDeep:{el | if ( stream_<el> != NULL) stream_<el>->reset(stream_<el>);<\n>}>
}<\n>
>>

rewriteClosureBlock(
 alt,
 rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,  // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
{
 while ( <referencedElements:{el | (stream_<el> != NULL && stream_<el>->hasNext(stream_<el>))
 }; separator="|| "> )
 {
  <alt>
 }
 <referencedElements:{el | if (stream_<el> != NULL) stream_<el>->reset(stream_<el>);<\n>}>
}<\n>
>>
RewriteEarlyExitException() ::=
<<
CONSTRUCTEX();
EXCEPTION->type         = ANTLR3_REWRITE_EARLY_EXCEPTION;
EXCEPTION->name         = (void *)ANTLR3_REWRITE_EARLY_EXCEPTION_NAME;
>>
rewritePositiveClosureBlock(
 alt,
 rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,  // elements in immediately block; no nested blocks
 description) ::=
<<
if (<referencedElements:{el | (stream_<el> == NULL || !stream_<el>->hasNext(stream_<el>)) }; separator="|| "> )
{
   <RewriteEarlyExitException()>
}
else
{
 while ( <referencedElements:{el | (stream_<el>->hasNext(stream_<el>)) }; separator="|| "> ) {
  <alt>
 }
 <referencedElements:{el | stream_<el>->reset(stream_<el>);<\n>}>
}
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
   <a.alt>
}<\n>
<else>
{
   <a.alt>
```

```
}<\n>
<endif>
>>

/** For empty
 rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = NULL; /* \<-- rewriteEmptyAlt()) */"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
   <ASTLabelType> root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->nilNode(ADAPTOR));
   <root:rewriteElement()>
   <children:rewriteElement()>
   ADAPTOR->addChild(ADAPTOR, root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
<endif>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextNode(stream_<label>));<\n>
>>

/** Gen $label ... where defined via label+=ID
 */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextNode(stream_<label>));<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRootToken(ADAPTOR, stream_<label> == NULL ?
NULL : stream_<label>->nextToken(stream_<label>), root_<treeLevel>));<\n>
```

```
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR,
<createRewriteNodeFromElement(...)>, root_<treeLevel>));<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
root_<treeLevel>
 = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>));<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule,dup) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<rule> == NULL ? NULL : stream_<rule>-
>nextTree(stream_<rule>));<\n>
>>

rewriteRuleRefRoot(rule,dup) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, stream_<rule> == NULL ? NULL
: stream_<rule>->nextNode(stream_<rule>), root_<treeLevel>));<\n>
>>

rewriteNodeAction(action) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action)
```

```
 ::= <<
root_<treeLevel> = (<ASLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, <action>, root_<treeLevel>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
ADAPTOR->addChild(ADAPTOR, root_<treeLevel>, stream_<label> == NULL ? NULL : stream_<label>-
>nextTree(stream_<label>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot(ADAPTOR, stream_<label> == NULL ? NULL
: stream_<label>->nextNode(stream_<label>), root_<treeLevel>));<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)(ADAPTOR->becomeRoot((<ASTLabelType>)(stream_<label>
 == NULL ? NULL : stream_<label>->nextNode(stream_<label>), root_<treeLevel>));<\n>
>>

createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel, args) !>
<hetero>New(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
<if(args)>

#if <length(args)> == 2
 (<ASTLabelType>)ADAPTOR->createTypeTokenText(ADAPTOR, <tokenType>, TOKTEXT(<args; separator=",
">))
#else
 (<ASTLabelType>)ADAPTOR->createTypeText(ADAPTOR, <tokenType>, (pANTLR3_UINT8)<args;
separator=", ">)
#endif

<else>
(<ASTLabelType>)ADAPTOR->createTypeText(ADAPTOR, <tokenType>, (pANTLR3_UINT8)"<tokenType>")
<endif>
<endif>
>>
```

```
createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
<hetero>New(stream_<token>->nextToken(stream_<token>)<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>

#if <length(args)> == 2
ADAPTOR->createTypeTokenText(ADAPTOR, <token>->getType(<token>, TOKTEXT(<token>, <args;
separator=",
 ">)) /* JIMI */
#else
ADAPTOR->createTypeToken(ADAPTOR, <token>->getType(<token>, <token>, <args; separator=", ">)
#endif

<else>
stream_<token> == NULL ? NULL : stream_<token>->nextNode(stream_<token>)
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/C/AST.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during normal parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  The situation is not too bad as rewrite (->) usage makes ^ and !
*  invalid. There is no huge explosion of
 combinations.
*/
group ASTParser;

@rule.setErrorReturnValue() ::= <<
RetVal.Tree := Adaptor.ErrorNode(Input, RetVal.Start as IToken,
 Input.LT(-1), RE) as I<ASTLabelType>;
>>

// TOKEN AST STUFF

/** ID and output=AST */
tokenRef(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
<label>_tree := <createNodeFromToken(...)>;
Adaptor.AddChild(Root[0], <label>_tree);
<if(backtracking)>
end;
<endif>
>>

/** ID! and output=AST (same as plain tokenRef) */
tokenRefBang(token,label,elementIndex) ::= "<super.tokenRef(...)>"
```

```
/** ID^ and output=AST */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
<super.tokenRef(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
<label>_tree := <createNodeFromToken(...)>;
Root[0] := Adaptor.BecomeRoot(<label>_tree, Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
>>

/** ids+=ID! and output=AST */
tokenRefBangAndListLabel(token,label,elementIndex,hetero)
 ::= <<
<tokenRefBang(...)>
<listLabel(elem=label,...)>
>>

/** label+=TOKEN when output=AST but not rewrite alt */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** Match label+=TOKEN^ when output=AST but not rewrite alt */
tokenRefRuleRootAndListLabel(token,label,hetero,elementIndex) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

// the match set stuff is interesting in that it uses an argument list
// to pass code to the default matchSet; another possible way to alter
// inherited code.  I don't use the region stuff because I need to pass
// different chunks depending on the operator.  I don't like making
// the template name have the operator as the number of templates gets
// large but this is the most flexible--this is as opposed to having
// the code generator call matchSet then add root code or ruleroot code
// plus list label plus ...  The combinations might require complicated
//
 rather than just added on code.  Investigate that refactoring when
// I have more time.
```

```
matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
<super.matchSet(..., postmatchCode={<if(backtracking)>if (State.Backtracking = 0) then
<endif>Adaptor.AddChild(Root[0], <createNodeFromToken(...)>);})>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= "<super.matchSet(...)>"

// note there is no matchSetTrack because -> rewrites force sets to be
// plain old blocks of alts: (A|B|...|C)

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<if(label)>
<label> := Input.LT(1) as I<labelType>;<\n>
<endif>
<super.matchSet(..., postmatchCode={<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[0] :=
Adaptor.BecomeRoot(<createNodeFromToken(...)>, Root[0]) as I<ASTLabelType>;})>
>>

// RULE REF AST

/** rule when output=AST */
ruleRef(rule,label,elementIndex,args,scope)
 ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Adaptor.AddChild(Root[0], <label>.Tree);
>>

/** rule! is same as normal rule ref */
ruleRefBang(rule,label,elementIndex,args,scope) ::= "<super.ruleRef(...)>"

/** rule^ */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>Root[0] := Adaptor.BecomeRoot(<label>.Tree, Root[0])
as I<ASTLabelType>;
>>

/** x+=rule when output=AST */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule! when output=AST is a rule ref with list addition */
```

```
ruleRefBangAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefBang(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** x+=rule^ */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".Tree",...)>
>>

// WILDCARD AST

wildcard(label,elementIndex)
 ::= <<
<super.wildcard(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
 <label>_tree := Adaptor.CreateNode(<label>) as I<ASTLabelType>;
 Adaptor.AddChild(Root[0], <label>_tree);
<if(backtracking)>
end;
<endif>
>>

wildcardBang(label,elementIndex) ::= "<super.wildcard(...)>"

wildcardRuleRoot(label,elementIndex) ::= <<
<super.wildcard(...)>
<if(backtracking)>
if (State.Backtracking = 0) then
begin
<endif>
 <label>_tree := Adaptor.CreateNode(<label>) as I<ASTLabelType>;
 Root[0] := Adaptor.BecomeRoot(<label>_tree, Root[0]) as I<ASTLabelType>;
<if(backtracking)>
end;
<endif>
>>

createNodeFromToken(label,hetero) ::= <<
<if(hetero)>
T<hetero>.Create(<label>) <! new MethodNode(IDLabel) !>
<else>
Adaptor.CreateNode(<label>) as I<ASTLabelType>
<endif>
```

```
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
RetVal.Tree := Adaptor.RulePostProcessing(Root[0]) as I<ASTLabelType>;
<if(!TREE_PARSER)>
Adaptor.SetTokenBoundaries(RetVal.Tree,
 RetVal.Start as IToken, RetVal.Stop as IToken);
<endif>
<if(backtracking)>
<\n>end;
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Delphi/ASTParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

```
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : C# by code generator.
*/
group ASTDbg;

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor;
public ITreeAdaptor TreeAdaptor
{
 get {
<if(grammar.grammarIsRoot)>
  return this.adaptor;
<else>
  this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
      <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
 set { this.adaptor = new DebugTreeAdaptor(dbg, value); }
}<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy = new DebugEventSocketProxy(this, port, adaptor);
DebugListener = proxy;
<!
Original line follows, replaced by the next two ifs:
set<inputStreamType>(new Debug<inputStreamType>(input,proxy));
!>
<if(PARSER)>
TokenStream = new DebugTokenStream(input,proxy);<\n>
<endif>
<if(TREE_PARSER)>
TokenStream = new DebugTreeNodeStream(input,proxy);<\n>
<endif>
try {
   proxy.Handshake();
} catch (IOException ioe) {
   ReportError(ioe);
```

```
}
>>

@ctorForRootGrammar.finally() ::= <<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

@ctorForProfilingRootGrammar.finally() ::=<<
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;
proxy.TreeAdaptor = adap;
>>

@ctorForPredefinedListener.superClassRef()
 ::= "base(input, dbg)"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>
ITreeAdaptor adap = new CommonTreeAdaptor();
TreeAdaptor = adap;<\n>
<endif>
>>

@rewriteElement.pregen() ::= "dbg.Location(<e.line>,<e.pos>);"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products

   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
using Antlr.Runtime.Tree;<\n>
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected ITreeAdaptor adaptor = new CommonTreeAdaptor();<\n>
public ITreeAdaptor TreeAdaptor
{
    get { return this.adaptor; }
   set {
 this.adaptor = value;
 <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor = this.adaptor;}>
 }
}
>>

@returnScope.ruleReturnMembers() ::= <<
private <ASTLabelType> tree;
override public object Tree
{
 get { return tree; }
 set { tree = (<ASTLabelType>) value; }
}
```

```
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
 ruleDescriptor.wildcardTreeListLabels]:{<ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{RewriteRule<rewriteElementType>Stream stream_<it> = new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
    :{RewriteRuleSubtreeStream stream_<it> = new RewriteRuleSubtreeStream(adaptor,"rule <it>");};
separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)adaptor.GetNilNode();<\n>
<endif>
<endif>
<endif>
>>

// T r a c k i n g  R u l e  E l e m e n t s

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.Add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,hetero)
```

```
 ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.Add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.Add(<label>.Tree);
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::=
 <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements:          <referencedElementsDeep; separator=", ">
// token labels:      <referencedTokenLabels; separator=", ">
// rule labels:       <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels:  <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels];
 separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {<\n>
<endif>
<prevRuleRootRef()>.Tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.GetNilNode();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.Tree = (<ASTLabelType>)adaptor.RulePostProcessing(root_0);
input.ReplaceChildren(adaptor.GetParent(retval.Start),
                  adaptor.GetChildIndex(retval.Start),
                  adaptor.GetChildIndex(_last),
                  retval.Tree);
<endif>
<endif>
<! if parser or rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.Tree = root_0;
<endif>
<if(!rewriteMode)>
<prevRuleRootRef()>.Tree = root_0;
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{RewriteRule<rewriteElementType>Stream stream_<it> = new
```

```
RewriteRule<rewriteElementType>Stream(adaptor,
 "token <it>", <it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{RewriteRule<rewriteElementType>Stream stream_<it> = new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
   separator="\n"
>
<referencedWildcardLabels
   :{RewriteRuleSubtreeStream stream_<it> = new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
   separator="\n"
>
<referencedWildcardListLabels
   :{RewriteRuleSubtreeStream stream_<it> = new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",list_<it>);};
   separator="\n"
>
<referencedRuleLabels
   :{RewriteRuleSubtreeStream stream_<it> = new RewriteRuleSubtreeStream(adaptor, "rule <it>", <it>!=null ?
<it>.Tree : null);};
   separator="\n"
>
<referencedRuleListLabels
   :{RewriteRuleSubtreeStream stream_<it> = new RewriteRuleSubtreeStream(adaptor, "token <it>", list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *
  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.HasNext()}; separator=" || "> )
{
   <alt>
}
<referencedElementsDeep:{el | stream_<el>.Reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.HasNext()}; separator=" || "> )
{
   <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el
 | stream_<el>.HasNext()}; separator=" || ">) ) {
   throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el | stream_<el>.HasNext()}; separator=" || "> )
{
   <alt>
}
<referencedElements:{el | stream_<el>.Reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>)
{
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
```

```
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.GetNilNode();
<root:rewriteElement()>
<children:rewriteElement()>
adaptor.AddChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
adaptor.AddChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/**
 Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
adaptor.AddChild(root_<treeLevel>, <createImaginaryNode(tokenType=token,
 ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>,
root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.Tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<rule>.NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<rule>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.AddChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel>
 = (<ASTLabelType>)adaptor.BecomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
```

```
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.BecomeRoot(stream_<label>.NextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.AddChild(root_<treeLevel>, stream_<label>.NextTree());<\n>
>>

createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel,
 args) !>
new <hetero>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.Create(<tokenType>, <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.NextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.Create(<token>, <args; separator=", ">)
<else>
stream_<token>.NextNode()
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp2/AST.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp/AST.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/ObjCTarget.java
No license file was found, but licenses were detected in source scan.

```
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
/** Template overrides to add debugging to normal Python output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
group Dbg;

@outputFile.imports() ::= <<
<@super.imports()>
from antlr3.debug import *
>>

@genericParser.args() ::= <<
debug_socket = kwargs.pop('debug_socket', None)
port = kwargs.pop('port', None)
>>

@genericParser.init() ::= <<
self.ruleLevel = 0

if self._dbg is None:
   <createListenerAndHandshake()>

>>

createListenerAndHandshake() ::= <<
<if(TREE_PARSER)>
proxy
 = DebugEventSocketProxy(self, adaptor=self.input.getTreeAdaptor(),
                          debug=debug_socket, port=port)<\n>
<else>
proxy = DebugEventSocketProxy(self, debug=debug_socket, port=port)<\n>
<endif>
self.setDebugListener(proxy)
proxy.handshake()

>>

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
ruleNames = [
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n    ", separator=", ">
  ]<\n>
<endif>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s) !>
def getRuleLevel(self):
  return self.ruleLevel

def incRuleLevel(self):
  self.ruleLevel += 1

def decRuleLevel(self):
  self.ruleLevel -= 1

<if(profile)>
  <ctorForProfilingRootGrammar()>
<else>
  <ctorForRootGrammar()>
<endif>
<ctorForPredefinedListener()>
<else> <! imported grammar !>
def getRuleLevel(self):
  return <grammar.delegators:{g| <g:delegateName()>}>.getRuleLevel()

def incRuleLevel(self):
  <grammar.delegators:{g| <g:delegateName()>}>.incRuleLevel()

def
 decRuleLevel(self):
  <grammar.delegators:{g| <g:delegateName()>}>.decRuleLevel()

<ctorForDelegateGrammar()>
<endif>
<if(profile)>
FIXME(2)
public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
  ((Profiler)self._dbg).examineRuleMemoization(input, ruleIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
  return super.alreadyParsedRule(input, ruleIndex);
}<\n>
FIXME(3)
public void memoize(IntStream input,
            int ruleIndex,
            int ruleStartIndex)
{
  ((Profiler)self._dbg).memoize(input, ruleIndex, ruleStartIndex,
<grammar.composite.rootGrammar.recognizerName>.ruleNames[ruleIndex]);
  super.memoize(input, ruleIndex, ruleStartIndex);
}<\n>
```

```
<endif>
def evalPredicate(self, result, predicate):
  self._dbg.semanticPredicate(result, predicate)
  return result
<\n>
>>

ctorForRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
<! Same except we
 add port number and profile stuff if root grammar !>
<!
public <name>(<inputStreamType> input) {
  this(input, DebugEventSocketProxy.DEFAULT_DEBUGGER_PORT, new RecognizerSharedState());
}
public <name>(<inputStreamType> input, int port, RecognizerSharedState state) {
  super(input, state);
  <parserCtorBody()>
  <createListenerAndHandshake()>
  <grammar.directDelegates:{g|<g:delegateName()> = new <g.recognizerName>(input, self._dbg, this.state, this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
  <@finally()>
}<\n>
!>
>>

ctorForProfilingRootGrammar() ::= <<
<! bug: can't use <@super.members()> cut-n-paste instead !>
public <name>(<inputStreamType> input) {
  this(input, new Profiler(null), new RecognizerSharedState());
}
public <name>(<inputStreamType> input, DebugEventListener self.dbg, RecognizerSharedState state) {
  super(input, self.dbg, state);
  Profiler p = (Profiler)self.dbg;
  p.setParser(this);
  <parserCtorBody()>
  <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, self.dbg, this.state, this<grammar.delegators:{g|, <g:delegateName()>}>);}; separator="\n">
  <@finally()>
}
<\n>
>>

/** Basically we don't want to set any dbg listeners are root will have it. */
ctorForDelegateGrammar() ::= <<
<!
public <name>(<inputStreamType> input, DebugEventListener self.dbg, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}>) {
   super(input, dbg, state);
   <parserCtorBody()>
   <grammar.directDelegates:
    {g|<g:delegateName()> = new <g.recognizerName>(input, this, this.state<grammar.delegators:{g|,
<g:delegateName()>}>);}; separator="\n">
}<\n>
!>
>>

ctorForPredefinedListener() ::= <<
<!
public <name>(<inputStreamType> input, DebugEventListener dbg) {
   <@superClassRef>super(input, dbg, new RecognizerSharedState());<@end>
<if(profile)>
   Profiler p = (Profiler)dbg;
   p.setParser(this);
<endif>
   <parserCtorBody()>
   <grammar.directDelegates:{g|<g:delegateName()>
 = new <g.recognizerName>(input, self._dbg, this.state, this<grammar.delegators:{g|, <g:delegateName()>}>);};
separator="\n">
   <@finally()>
}<\n>
!>
>>

@genericParser.superClassName() ::= "Debug<@super.superClassName()>"

@rule.body() ::= <<
try:
   self._dbg.enterRule(self.getGrammarFileName(), "<ruleName>")
   if self.getRuleLevel() == 0:
       self._dbg.commence();
   self.incRuleLevel()
   self._dbg.location(<ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>)

   <@super.body()>

   self._dbg.location(<ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>)
finally:
   self._dbg.exitRule(self.getGrammarFileName(), "<ruleName>")
   self.decRuleLevel()
   if self.getRuleLevel() == 0:
        self._dbg.terminate()

>>
```

```
@synpred.start() ::= "self._dbg.beginBacktrack(self._state.backtracking)"

@synpred.stop() ::= "self._dbg.endBacktrack(self._state.backtracking, success)"

// Common debug event triggers used by region overrides
 below

enterSubRule() ::=
   "try { self._dbg.enterSubRule(<decisionNumber>);<\n>"

exitSubRule() ::=
   "} finally {self._dbg.exitSubRule(<decisionNumber>);}<\n>"

enterDecision() ::=
   "try { self._dbg.enterDecision(<decisionNumber>);<\n>"

exitDecision() ::=
   "} finally {self._dbg.exitDecision(<decisionNumber>);}<\n>"

enterAlt(n) ::= "self._dbg.enterAlt(<n>)<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.body() ::= <<
try:
   self._dbg.enterSubRule(<decisionNumber>)
   <@super.body()>
finally:
   self._dbg.exitSubRule(<decisionNumber>)
>>

@blockBody.decision() ::= <<
try:
   self._dbg.enterDecision(<decisionNumber>)
   <@super.decision()>
finally:
   self._dbg.exitDecision(<decisionNumber>)
>>

@ruleBlock.decision() ::= <<
try:
   self._dbg.enterDecision(<decisionNumber>)
   <@super.decision()>
finally:
   self._dbg.exitDecision(<decisionNumber>)
>>

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"
```

```
@blockSingleAlt.prealt()
 ::= "<enterAlt(n=\"1\")>"

@positiveClosureBlock.loopBody() ::= <<
try:
  self._dbg.enterSubRule(<decisionNumber>)
  <@super.loopBody()>
finally:
  self._dbg.exitSubRule(<decisionNumber>)<\n>
>>

@positiveClosureBlockLoop.decisionBody() ::= <<
try:
  self._dbg.enterDecision(<decisionNumber>)
  <@super.decisionBody()>
finally:
  self._dbg.exitDecision(<decisionNumber>)
>>

@positiveClosureBlockLoop.earlyExitException() ::=
  "self._dbg.recognitionException(eee)<\n>"

@closureBlock.loopBody() ::= <<
try:
  self._dbg.enterSubRule(<decisionNumber>)
  <@super.loopBody()>
finally:
  self._dbg.exitSubRule(<decisionNumber>)<\n>
>>

@closureBlockLoop.decisionBody() ::= <<
try:
  self._dbg.enterDecision(<decisionNumber>)
  <@super.decisionBody()>
finally:
  self._dbg.exitDecision(<decisionNumber>)
>>

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
  "self._dbg.location(<it.line>, <it.pos>)"

@matchSet.mismatchedSetException() ::=

  "self._dbg.recognitionException(mse)"
```

```
@dfaState.noViableAltException() ::= "self._dbg.recognitionException(nvae)"

@dfaStateSwitch.noViableAltException() ::= "self._dbg.recognitionException(nvae)"

dfaDecision(decisionNumber,description) ::= <<
try:
    self.isCyclicDecision = True
    <super.dfaDecision(...)>

except NoViableAltException, nvae:
    self._dbg.recognitionException(nvae)
    raise

>>

@cyclicDFA.errorMethod() ::= <<
def error(self, nvae):
    self._dbg.recognitionException(nvae)

>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
self.evalPredicate(<pred>,"<description>")
>>
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/Dbg.stg

No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2006,2007 Kay Roepke
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR

```
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

/*
 Template group file for the Objective C code generator.
 Heavily based on Java.stg

 Written by Kay Roepke <kroepke(at)classdump.org>

 This file is part of ANTLR and subject to the same license as ANTLR itself.
*/

group ObjC implements ANTLRCore;

objcTypeInitMap ::= [
 "int":"0",
 "long":"0",
 "float":"0.0",
 "double":"0.0",
 "boolean":"false",
 "byte":"0",
 "short":"0",
 "char":"0",
 "id":"nil",
 default:"nil" // anything other than an atomic type
]

className() ::= "<name><!
 if(LEXER)>Lexer<else><if(TREE_PARSER)>Tree<endif>Parser<endif !>"

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass,literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
#import "<name><!if(LEXER)>Lexer<else><if(TREE_PARSER)>Tree<endif>Parser<endif!>.h"
<@end>

<docComment>
<recognizer>
>>



headerFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName,
 ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass,literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
#import \<Cocoa/Cocoa.h>
#import \<ANTLR/ANTLR.h>
<@end>

<actions.(actionScope).header>

<if(LEXER)>
<lexerHeaderFile(...)>
<endif>
<if(PARSER)>
<parserHeaderFile(...)>
<endif>
<if(TREE_PARSER)>
<treeParserHeaderFile(...)>
<endif>
>>

lexerHeaderFile() ::=
<<
<cyclicDFAs:cyclicDFAInterface()>

#pragma mark Rule return scopes start
<rules:{rule |
<rule:{ruleDescriptor | <returnScopeInterface(scope=ruleDescriptor.returnScope)>}>
}>
#pragma mark Rule return scopes end

#pragma mark Tokens
<tokens:{#define <name>_<it.name> <it.type>}; separator="\n">

@interface <className()> : ANTLRLexer {
 <cyclicDFAs:{dfa | <name>DFA<dfa.decisionNumber> *dfa<dfa.decisionNumber>;}; separator="\n">
 <synpreds:{p | SEL <p>SyntacticPredicate;}; separator="\n">
 <actions.lexer.ivars>
}

<actions.lexer.methodsdecl>

<rules:{rule |
- (<rule.ruleDescriptor:{ruleDescriptor|<returnType()>}>)
<if(!rule.ruleDescriptor.isSynPred)>m<endif><rule.ruleName><if(rule.ruleDescriptor.parameterScope)><rule.ruleDescriptor.parameterScope:parameterScope(scope=it)><endif>;
}>


@end
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="id<ANTLRToken> ",
     filterMode) ::= <<
<cyclicDFAs:cyclicDFA()>

/** As per Terence: No returns for lexer rules!
#pragma mark Rule return scopes start
<rules:{rule |
<rule.ruleDescriptor:{ruleDescriptor | <returnScope(scope=ruleDescriptor.returnScope)>}>
}>
#pragma mark Rule return scopes end
*/
@implementation <className()>

static NSArray *tokenNames;

<actions.lexer.methods>

+ (void) initialize
{
   // todo: get tokenNames into lexer - requires changes to CodeGenerator.java and ANTLRCore.sti
   tokenNames = [[NSArray alloc] init];
}
```

```
- (id) initWithCharStream:(id\<ANTLRCharStream>)anInput
{
 if (nil!=(self = [super initWithCharStream:anInput])) {
  <if(memoize)>
  // initialize the memoization cache - the
 indices are 1-based in the runtime code!
  [ruleMemo addObject:[NSNull null]];     // dummy entry to ensure 1-basedness.
  for (int i = 0; i \< <numRules>; i++) {
      [[state ruleMemo] addObject:[NSMutableDictionary dictionary]];
  }
  <endif>
  <synpreds:{p | <lexerSynpred(name=p)>};separator="\n">
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = [[<name>DFA<dfa.decisionNumber> alloc]
initWithRecognizer:self];}; separator="\n">
  <actions.lexer.init>
 }
 return self;
}

- (void) dealloc
{
 <cyclicDFAs:{dfa | [dfa<dfa.decisionNumber> release];}; separator="\n">
<actions.lexer.dealloc>
 [super dealloc];
}

+ (NSString *) tokenNameForType:(int)aTokenType
{
   return nil;
}

+ (NSArray *) tokenNames
{
   return tokenNames;
}

- (NSString *) grammarFileName
{
 return @"<fileName>";
}

<if(actions.lexer.reset)>
- (void) reset
{
<actions.lexer.reset>
 [super reset];
}
<endif>
```

```
<if(filterMode)>
<filteringNextToken()>
<endif>

<rules; separator="\n\n">

@end
>>

/** A override of Lexer.nextToken()
 that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*/
filteringNextToken() ::= <<
- (id\<ANTLRToken>) nextToken
{
   <sharedStateLocalVarDefinition()>
   while (YES) {
      if ( [input LA:1] == ANTLRCharStreamEOF ) {
         return nil; // should really be a +eofToken call here -> go figure
      }
      [self setToken:nil];
      [_state setChannel:ANTLRTokenChannelDefault];
      [_state setTokenStartLine:[input line]];
      [_state setTokenCharPositionInLine:[input charPositionInLine]];
      [_state setTokenStartCharIndex:[self charIndex]];
      @try {
         int m = [input mark];
         [_state setBacktracking:1];
         [_state setIsFailed:NO];
         [self mTokens];
         [_state setBacktracking:0];
         if ( [_state isFailed] ) {
            [input rewind:m];

   [input consume]; <! // advance one char and try again !>
         } else {
            [self emit];
            return token;
         }
      }
      @catch (ANTLRRecognitionException *re) {
         // shouldn't happen in backtracking mode, but...
         [self reportError:re];
         [self recover:re];
      }
```

```
  }
}
>>

filteringActionGate() ::= "[_state backtracking] == 1"

treeParserHeaderFile(LEXER, PARSER, TREE_PARSER, actionScope, actions, docComment,
         recognizer, name, tokens, tokenNames, rules, cyclicDFAs,
         bitsets, buildTemplate, profile, backtracking, synpreds,
         memoize, numRules, fileName, ANTLRVersion, generatedTimestamp, trace, scopes,
superClass="ANTLRTreeParser") ::= <<
<parserHeaderFile(...)>
>>

parserHeaderFile(LEXER, PARSER, TREE_PARSER, actionScope, actions, docComment,
         recognizer, name, tokens, tokenNames, rules, cyclicDFAs,
         bitsets, buildTemplate, profile, backtracking, synpreds,
          memoize, numRules, fileName, ANTLRVersion, generatedTimestamp, trace, scopes,
superClass="ANTLRParser") ::=
<<

<cyclicDFAs:cyclicDFAInterface()>

#pragma mark Tokens
<tokens:{#define <name>_<it.name> <it.type>}; separator="\n">

#pragma mark Dynamic Global Scopes
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeDecl(scope=it)><endif>}>

#pragma mark Dynamic Rule Scopes
<rules:{rule |
<rule.ruleDescriptor:{ ruleDescriptor | <ruleAttributeScopeDecl(scope=ruleDescriptor.ruleScope)>}>
}>

#pragma mark Rule Return Scopes
<rules:{rule |
<rule.ruleDescriptor:{ ruleDescriptor | <returnScopeInterface(scope=ruleDescriptor.returnScope)>}>
}>


@interface <className()> : <@superClassName><superClass><@end> {

 <cyclicDFAs:{dfa | <name>DFA<dfa.decisionNumber> *dfa<dfa.decisionNumber>;}; separator="\n">
 <synpreds:{p | SEL <p>SyntacticPredicate;}; separator="\n">
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeIVar(scope=it)><endif>}>
 <rules:{rule |
 <rule.ruleDescriptor.ruleScope:ruleAttributeScopeIVar(scope=it)>
 }>
 <@ivars()>

 <actions.(actionScope).ivars>
 }

<actions.(actionScope).methodsdecl>

<rules:{rule |
- (<rule.ruleDescriptor:{ruleDescriptor|<returnType()>}>)
<rule.ruleName><if(rule.ruleDescriptor.parameterScope)><rule.ruleDescriptor.parameterScope:parameterScope(scope=it)><endif>;
}>

<@methodsdecl()>

@end
>>

/** How to generate a parser */
genericParser(name, scopes, tokens, tokenNames, rules, numRules,
	cyclicDFAs, bitsets, inputStreamType, superClass,
	ASTLabelType="id", labelType, members) ::= <<

<cyclicDFAs:cyclicDFA()>

#pragma mark Bitsets
<bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>

#pragma mark Dynamic Global Scopes
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeImpl(scope=it)><endif>}>

#pragma mark Dynamic Rule Scopes
<rules:{rule |
<rule.ruleDescriptor.ruleScope:ruleAttributeScopeImpl(scope=it)>
}>

#pragma mark Rule return scopes
 start
<rules:{rule |
<rule.ruleDescriptor:{ruleDescriptor | <returnScope(scope=ruleDescriptor.returnScope)>}>
}>

@implementation <className()>

static NSArray *tokenNames;

+ (void) initialize
{
 <bitsets:bitsetInit(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>}, words64=it.bits)>
 tokenNames = [[NSArray alloc] initWithObjects:@"\<invalid>", @"\<EOR>", @"\<DOWN>", @"\<UP>",
<tokenNames:{@<it>}; separator=", ", wrap="\n ">, nil];
}

<if(PARSER)>
- (id) initWithTokenStream:(<inputStreamType>)aStream
{
 if ((self = [super initWithTokenStream:aStream])) {
<else><!TREE_PARSER!>
- (id) initWithTreeNodeStream:(<inputStreamType>)aStream
{
 if ((self = [super initWithTreeNodeStream:aStream])) {
<endif><\n>
<if(memoize)>
  ruleMemo = [[NSMutableDictionary alloc] initWithCapacity:<numRules>+1];
<endif>
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = [[<name>DFA<dfa.decisionNumber> alloc]
initWithRecognizer:self];}; separator="\n">
  <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeInit(scope=it)><endif>}>
  <rules:{rule
 |
  <rule.ruleDescriptor.ruleScope:ruleAttributeScopeInit(scope=it)>
  }>
  <actions.(actionScope).init>
  <@init()>
 }
 return self;
}

- (void) dealloc
{
<if(memoize)>
 [ruleMemo release];
<endif>
 <cyclicDFAs:{dfa | [dfa<dfa.decisionNumber> release];}; separator="\n">
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeDealloc(scope=it)><endif>}>
 <actions.(actionScope).dealloc>
 <@dealloc()>
 [super dealloc];
}

- (NSString *) grammarFileName
{
 return @"<fileName>";
}

<actions.(actionScope).methods>
```

```
<rules; separator="\n\n">

<synpreds:{p | <synpred(p)>}>

<@methods()>

@end
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets, ASTLabelType,
superClass="ANTLRParser", labelType="id<ANTLRToken> ", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="id\<ANTLRTokenStream>", ...)>
>>

treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>},
 ASTLabelType="id", superClass="ANTLRTreeParser", members={<actions.treeparser.members>}) ::= <<
<genericParser(inputStreamType="id\<ANTLRTreeNodeStream>", ...)>
>>

/** Maintain a local variable for the shared state object to avoid calling the accessor all the time. */
sharedStateLocalVarDefinition() ::= <<
<if(LEXER)>ANTLRLexerState<else>ANTLRBaseRecognizerState<endif> *_state = [self state];
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
- (void) <ruleName>
{
   <sharedStateLocalVarDefinition()>
   <if(trace)>NSLog(@"enter <ruleName>
 %d failed=%@ backtracking=%d", [input LA:1], [_state isFailed] ? @"YES" : @"NO", [_state backtracking]);
   @try {
       <block>
   }
   @finally {
       NSLog(@"exit <ruleName> %d failed=%@ backtracking=%d", [input LA:1], [_state isFailed] ? @"YES" :
@"NO", [_state backtracking]);
   }
<else>
```

```
   <block>
<endif>
}
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if ([_state isFailed]) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if ([_state isBacktracking]) {[_state setIsFailed:YES]; return <ruleReturnValue()>;}<endif>
>>

synpred(name) ::= <<
<!name>SyntacticPredicate = @selector(<name>); !>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ([_state isBacktracking] && [self alreadyParsedRule:input ruleIndex:<ruleDescriptor.index>] ) { return
<ruleReturnValue()>;
 }
<endif>
>>


/** How to generate code for a rule.
*  The return type aggregates are declared in the header file (headerFile template)
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
// $ANTLR start <ruleName>
// <fileName>:<description>
- (<returnType()>) <ruleName><ruleDescriptor.parameterScope:parameterScope(scope=it)>
{
   <if(trace)>NSLog(@"enter <ruleName> %@ failed=%@ backtracking=%d", [input LT:1], [_state isFailed] ?
@"YES" : @"NO", [_state backtracking]);<endif>
   <sharedStateLocalVarDefinition()>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   <ruleMemoization(name=ruleName)>
   @try {
       <block>
   }
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
 <actions.(actionScope).rulecatch>
<else>
 @catch (ANTLRRecognitionException *re) {
  [self reportError:re];
  [self
 recover:input exception:re];
 }<\n>
<endif>
<endif>
<endif>
 @finally {
  <if(trace)>NSLog(@"exit <ruleName> %@ failed=%@ backtracking=%d", [input LT:1], [_state isFailed] ?
@"YES" : @"NO", [_state backtracking]);<endif>
  <ruleCleanUp()>
  <(ruleDescriptor.actions.finally):execAction()>
  <ruleScopeCleanUp()>
 }
 <@postamble()>
 return <ruleReturnValue()>;
}
// $ANTLR end <ruleName>
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
   <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> _retval = [[[<returnTypeName()> alloc] init] autorelease];
[_retval setStart:[input LT:1]];<\n>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<returnType()> _<ruleDescriptor.singleValueReturnName> = <initValue(typeName=returnType())>;
<endif>
<endif>
<if(memoize)>
```

```
int <ruleDescriptor.name>_StartIndex = [input index];
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{[<name>_<it>_stack addObject:[[[<name><it>Scope alloc]
 init] autorelease]];}; separator="\n">
<ruleDescriptor.ruleScope:{[<name>_<it.name>_stack addObject:[[[<name><it.name>Scope alloc] init]
autorelease]];}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{[<name>_<it>_stack removeLastObject];}; separator="\n">
<ruleDescriptor.ruleScope:{[<name>_<it.name>_stack removeLastObject];}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels]
   :{<labelType> _<it.label.text> = nil;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]
   :{NSMutableArray *_list_<it.label.text> = nil;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|ANTLRRuleReturnScope <ll.label.text> = nil;}; separator="\n">
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
_<ruleDescriptor.singleValueReturnName>
<else>
_retval
<endif>
<endif>
<endif>
>>

ruleCleanUp()
 ::= <<
// token+rule list labels
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels]:{[_list_<it.label.text> release];}; separator="\n">
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
[_retval setStop:[input LT:-1]];<\n>
<endif>
<endif>
```

```
<if(memoize)>
<if(backtracking)>
if ( [_state isBacktracking] ) { [self memoize:input ruleIndex:<ruleDescriptor.index>
startIndex:<ruleDescriptor.name>_StartIndex]; }
<endif>
<endif>
>>



/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block, memoize) ::= <<

- (void)
m<ruleName><if(ruleDescriptor.parameterScope)><ruleDescriptor.parameterScope:parameterScope(scope=it)><endif>
{
   <if(trace)>NSLog(@"enter <ruleName> %C line=%d:%d failed=%@ backtracking=%d", [input LA:1], [self line], [self charPositionInLine], [_state isFailed] ? @"YES" : @"NO", [_state backtracking]);<endif>

 <sharedStateLocalVarDefinition()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleMemoization(name=ruleName)>
   @try {
<ruleDescriptor.actions.init>
<if(nakedBlock)>
       <block><\n>
<else>
       int _type = <name>_<ruleName>;
       <block>
       [_state setTokenType:_type];<\n>
<endif>
   }
   @finally {
       <if(trace)>NSLog(@"exit <ruleName> %C line=%d:%d failed=%@ backtracking=%d", [input LA:1], [self line], [self charPositionInLine], [_state isFailed] ? @"YES" : @"NO", [_state backtracking]);<endif>
       // rule cleanup
       <ruleCleanUp()>
       <(ruleDescriptor.actions.finally):execAction()>
   }
   return;
}
// $ANTLR end <ruleName>
>>

/** How to generate code for the implicitly-defined lexer grammar rule
```

```
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
- (void) mTokens
{
   <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description)
 ::= <<
// <fileName>:<description> // block
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description> //ruleblock
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description> // ruleBlockSingleAlt
<decls>
<@prealt()>
<alts>
<@postalt()>
>>
```

```
/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description)
 ::= <<
// <fileName>:<description> // blockSingleAlt
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 0 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description> // positiveClosureBlock
int cnt<decisionNumber>=0;
<decls>
<@preloop()>

do {
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
     if ( cnt<decisionNumber> >= 1 )  goto loop<decisionNumber>;
     <if(backtracking)>
           <ruleBacktrackFailure()>
           <endif>
   ANTLREarlyExitException *eee = [ANTLREarlyExitException exceptionWithStream:input
decisionNumber:<decisionNumber>];
   <@earlyExitException()>
   @throw eee;
   }
   cnt<decisionNumber>++;
} while (YES); loop<decisionNumber>: ;
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)*
 block with 0 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
<decls>
<@preloop()>
```

```
do {
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
     goto loop<decisionNumber>;
   }
} while (YES); loop<decisionNumber>: ;
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> :
   <@prealt()>
   <it>
   break;<\n>
>>

/** An alternative is just a list of elements;
 at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel) ::= <<
// <fileName>:<description> // alt
{
<@declarations()>
<elements:element()>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
```

```
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)>
_<label>=(<labelType>)[input LT:1];<\n>
<endif>
[self match:input tokenType:<token> follow:FOLLOW_<token>_in_<ruleName><elementIndex>];
<checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (_list_<label> == nil)
 _list_<label> = [[NSMutableArray alloc] init];
[_list_<label> addObject:_<elem>];
>>

/** match a character
 */
charRef(char,label) ::= <<
<if(label)>
int _<label> = [input LA:1];<\n>
<endif>
[self matchChar:<char>];
<checkRuleBacktrackFailure()><\n>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
_<label> = [input LA:1];<\n>
<endif>
[self matchRangeFromChar:<a> to:<b>];<checkRuleBacktrackFailure()>
>>
```

```
/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
_<label> = (<labelType>)[input LT:1];<\n>
<endif>
if (<s>) {
 <postmatchCode>
 [input consume];
<if(!LEXER)>
 [_state setIsErrorRecovery:NO];
<endif>
 <if(backtracking)>[_state setIsFailed:NO];<endif>
} else {
 <ruleBacktrackFailure()>
 ANTLRMismatchedSetException *mse = [ANTLRMismatchedSetException exceptionWithSet:nil stream:input];
 <@mismatchedSetException()>
<if(LEXER)>
 [self recover:mse];
<else>
 [self recoverFromMismatchedSet:input exception:mse follow:FOLLOW_set_in_<ruleName><elementIndex>];
<endif>
 @throw mse;
}<\n>
>>

matchSetAndListLabel(s,label,elementIndex,postmatchCode)
 ::= <<
<matchSet(...)>
<listLabel(...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
int _<label>Start = [self charIndex];
[self matchString:<string>];
<checkRuleBacktrackFailure()>
_<label> = [[ANTLRCommonToken alloc] initWithInput:input tokenType:ANTLRTokenTypeInvalid
channel:ANTLRTokenChannelDefault start:_<label>Start stop:[self charIndex]];
<else>
[self matchString:<string>];
<checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
```

```
_<label> = (<labelType>)[input LT:1];<\n>
<endif>
[self matchAny:input];
<checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
int _<label> = [input LA:1];<\n>
<endif>
[self matchAny];
<checkRuleBacktrackFailure()><\n>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(...)>
>>

/**
 Match a rule reference by invoking it possibly with arguments
*  and a return value or values.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
[[_state following] addObject:FOLLOW_<rule.name>_in_<ruleName><elementIndex>];
<if(label)>
_<label> = [self <rule.name><if(args)>:<args; separator=" :"><endif>];<\n>
<else>
[self <rule.name><if(args)>:<args; separator=" :"><endif>];<\n>
<endif>
[[_state following] removeLastObject];
<checkRuleBacktrackFailure()><\n>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(...)>
>>

/** A lexer rule reference */
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
```

```
<if(label)>
int _<label>Start<elementIndex> = [self charIndex];
[self m<rule.name><if(args)>:<args; separator=" :"><endif>];
<checkRuleBacktrackFailure()><\n>
_<label> = [[ANTLRCommonToken alloc] initWithInput:input tokenType:ANTLRTokenTypeInvalid
channel:ANTLRTokenChannelDefault start:_<label>Start<elementIndex>
 stop:[self charIndex]-1];
[_<label> setLine:[self line]];
<else>
[self m<rule.name><if(args)>:<args; separator=" :"><endif>];
<checkRuleBacktrackFailure()><\n>
<endif>
>>

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int _<label>Start<elementIndex> = [self charIndex];
[self matchChar:ANTLRCharStreamEOF];
<checkRuleBacktrackFailure()><\n>
<labelType> _<label> = [[ANTLRCommonToken alloc] initWithInput:input tokenType:ANTLRTokenTypeEOF
channel:ANTLRTokenChannelDefault start:_<label>Start<elementIndex> stop:[self charIndex]-1];
[_<label> setLine:[self line]];
<else>
[self matchChar:ANTLRCharStreamEOF];
<checkRuleBacktrackFailure()><\n>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if
 ( [input LA:1] == ANTLRTokenTypeDOWN ) {
   [self match:input tokenType:ANTLRTokenTypeDOWN follow:nil]; <checkRuleBacktrackFailure()>
   <children:element()>
   [self match:input tokenType:ANTLRTokenTypeUP follow:nil]; <checkRuleBacktrackFailure()>
}
<else>
```

```
[self match:input tokenType:ANTLRTokenTypeDOWN follow:nil]; <checkRuleBacktrackFailure()>
<children:element()>
[self match:input tokenType:ANTLRTokenTypeUP follow:nil]; <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
   <ruleBacktrackFailure()>
   @throw [ANTLRFailedPredicateException exceptionWithRuleName:@"<ruleName>"
predicate:@"<description>" stream:input];
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
{
 int LA<decisionNumber>_<stateNumber> = [input LA:<k>];
 <edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
 alt<decisionNumber> = <eotPredictsAlt>;
<else>
 <ruleBacktrackFailure()>
   ANTLRNoViableAltException *nvae = [ANTLRNoViableAltException
exceptionWithDecision:<decisionNumber> state:<stateNumber> stream:input];
 <@noViableAltException()>
 @throw nvae;<\n>
<endif>
 }
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
 int LA<decisionNumber>_<stateNumber> = [input LA:<k>];
 <edges; separator="\nelse ">
}
```

```
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
 *  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
{
 int LA<decisionNumber>_<stateNumber> = [input LA:<k>];
 <edges; separator="\nelse "><\n>
 <if(eotPredictsAlt)>
 else {
     alt<decisionNumber> = <eotPredictsAlt>;
 }<\n>
 <endif>
}
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( <labelExpr> <if(predicates)>&& <predicates><endif>) {
 <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be
 generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>]) {
 <edges; separator="\n">
default:
<if(eotPredictsAlt)>
 alt<decisionNumber> = <eotPredictsAlt>;
<else> {
 <ruleBacktrackFailure()>
   ANTLRNoViableAltException *nvae = [ANTLRNoViableAltException
exceptionWithDecision:<decisionNumber> state:<stateNumber> stream:input];
 <@noViableAltException()>
 @throw nvae;<\n>
 }
<endif>
}
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>]) {
 <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ([input LA:<k>]) {
 <edges; separator="\n">
<if(eotPredictsAlt)>
default:
 alt<decisionNumber> = <eotPredictsAlt>
 break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case
 <it>:}; separator="\n">
 <targetState>
 break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = [dfa<decisionNumber> predict];
>>

/** Used in headerFile */
cyclicDFAInterface(dfa) ::= <<
#pragma mark Cyclic DFA interface start <name>DFA<dfa.decisionNumber>
@interface <name>DFA<dfa.decisionNumber> : ANTLRDFA {} @end<\n>
#pragma mark Cyclic DFA interface end <name>DFA<dfa.decisionNumber>
>>
```

```
/** Used in lexer/parser implementation files */
cyclicDFA(dfa) ::= <<
#pragma mark Cyclic DFA implementation start <name>DFA<dfa.decisionNumber>
@implementation <name>DFA<dfa.decisionNumber>
const static int <name>dfa<dfa.decisionNumber>_eot[<dfa.numberOfStates>] =
   {<dfa.eot; wrap="\n     ", separator=",", null="-1">};
const static int <name>dfa<dfa.decisionNumber>_eof[<dfa.numberOfStates>]
 =
   {<dfa.eof; wrap="\n     ", separator=",", null="-1">};
const static unichar <name>dfa<dfa.decisionNumber>_min[<dfa.numberOfStates>] =
   {<dfa.min; wrap="\n     ", separator=",", null="0">};
const static unichar <name>dfa<dfa.decisionNumber>_max[<dfa.numberOfStates>] =
   {<dfa.max; wrap="\n     ", separator=",", null="0">};
const static int <name>dfa<dfa.decisionNumber>_accept[<dfa.numberOfStates>] =
   {<dfa.accept; wrap="\n     ", separator=",", null="-1">};
const static int <name>dfa<dfa.decisionNumber>_special[<dfa.numberOfStates>] =
   {<dfa.special; wrap="\n     ", separator=",", null="-1">};
const static int <name>dfa<dfa.decisionNumber>_transition[] = {};
<dfa.edgeTransitionClassMap.keys:{ table |
const static int <name>dfa<dfa.decisionNumber>_transition<i0>[] = {<table; separator=", ", wrap="\n ", null="-
1">};
}; null="">

- (id) initWithRecognizer:(ANTLRBaseRecognizer *) theRecognizer
{
 if ((self = [super initWithRecognizer:theRecognizer])
 != nil) {
  eot = <name>dfa<dfa.decisionNumber>_eot;
  eof = <name>dfa<dfa.decisionNumber>_eof;
  min = <name>dfa<dfa.decisionNumber>_min;
  max = <name>dfa<dfa.decisionNumber>_max;
  accept = <name>dfa<dfa.decisionNumber>_accept;
  special = <name>dfa<dfa.decisionNumber>_special;
  if (!(transition = calloc(<dfa.numberOfStates>, sizeof(void*)))) {
   [self release];
   return nil;
  }
  <dfa.transitionEdgeTables:{whichTable|transition[<i0>] =
<name>dfa<dfa.decisionNumber>_transition<whichTable>;}; separator="\n", null="">
 }
 return self;
}

<if(dfa.specialStateSTs)>
- (int) specialStateTransition:(int) s
{
 int _s = s;
 switch ( s ) {
  <dfa.specialStateSTs:{state |
  case <i0> : <! compressed special state numbers 0..n-1 !>
  <state>}; separator="\n">
 }
<if(backtracking)>
 if ([recognizer isBacktracking]) {
  [recognizer setIsFailed:YES];
  return -1;
 }<\n>
<endif>
 ANTLRNoViableAltException *nvae = [ANTLRNoViableAltException
exceptionWithDecision:<dfa.decisionNumber>
 state:_s stream:[recognizer input]];
 <! [self error:nvae]; !> <! for debugger - do later !>
 @throw nvae;
}<\n>
<endif>

- (void) dealloc
{
 free(transition);
 [super dealloc];
}

- (NSString *) description
{
 return @"<dfa.description>";
}

<@errorMethod()>

@end
#pragma mark Cyclic DFA implementation end <name>DFA<dfa.decisionNumber>
<\n>
>>

cyclicDFAState(decisionNumber, stateNumber,edges,needErrorClause,semPredState) ::= <<
<if(semPredState)>
[[recognizer input] rewind];<\n>
<else>
int LA<decisionNumber>_<stateNumber> = [input LA:1];<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
if ( s>=0 ) return s;
break;
>>
```

```
/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump
 to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>

// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(first(operands)<rest(operands):{o | ||<o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "<pred>"

/** synpreds are b0rken in cyclic DFA special states
*  Damn! For now, work around with using the selectors directly, and by providing a trampoline evalSynPred
method in
*  ANTLRDFA
*/
/* evalSynPredicate(pred,description) ::= "[self evaluateSyntacticPredicate:<pred>SyntacticPredicate stream:input]"
*/
evalSynPredicate(pred,description) ::= "[self evaluateSyntacticPredicate:@selector(<pred>)]"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt)
 ::= "[input LA:<k>]==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber>\>=<lower> && LA<decisionNumber>_<stateNumber>\<=<upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "([input LA:<k>]\>=<lower>
&& [input LA:<k>]\<=<upper>)"

setTest(ranges) ::= "<ranges; separator=\"||\">"

// A T T R I B U T E S

globalAttributeScopeDecl(scope) ::= <<
@interface <name><scope.name>Scope : NSObject {
 <scope.attributes:{<it.decl>;}; separator="\n">
}
// use KVC to access attributes!
@end
>>

globalAttributeScopeIVar(scope) ::= <<
NSMutableArray *<name>_<scope.name>_stack;
>>

globalAttributeScopeImpl(scope) ::= <<
@implementation <name><scope.name>Scope
@end
>>

globalAttributeScopeInit(scope) ::= <<
<name>_<scope.name>_stack = [[NSMutableArray alloc] init];
>>

globalAttributeScopeDealloc(scope) ::= <<
[<name>_<scope.name>_stack release];
>>

ruleAttributeScopeDecl(scope)
 ::= <<
<if(scope.attributes)>
@interface <name><scope.name>Scope : NSObject {
 <scope.attributes:{<it.decl>;}; separator="\n">
}
// use KVC to access attributes!
@end
<endif>
>>

ruleAttributeScopeIVar(scope) ::= <<
NSMutableArray *<name>_<scope.name>_stack;
>>

ruleAttributeScopeImpl(scope) ::= <<
<if(scope.attributes)>
```

```
@implementation <name><scope.name>Scope
@end
<endif>
>>

ruleAttributeScopeInit(scope) ::= <<
<name>_<scope.name>_stack = [[NSMutableArray alloc] init];
>>

ruleAttributeScopeDealloc(scope) ::= <<
[<name>_<scope.name>_stack release];
>>



returnTypeName() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
<className()>_<ruleDescriptor.name>_return
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
<else>
void<! for synpreds this is always void !>
<endif>
>>

returnType() ::= <<
<returnTypeName()><if(!ruleDescriptor.isSynPred)><if(ruleDescriptor.hasMultipleReturnValues)>
 *<endif><endif>
>>

/** Generate the Objective-C type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<className()>_<referencedRule.name>_return *
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
```

```
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<objcTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> _<label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScopeInterface(scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
@interface
 <returnTypeName()> : ANTLR<if(TREE_PARSER)>Tree<endif>ParserRuleReturnScope {
   <scope.attributes:{<it.decl>;}; separator="\n">
   <@ivars()>
   <actions.(actionScope).ruleReturnIvars>
}
<@methods()>
<actions.(actionScope).ruleReturnMethodsDecl>
@end
<endif>
<endif>
>>

returnScope(scope) ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasMultipleReturnValues)>
@implementation <returnTypeName()>
<@methods()>
<actions.(actionScope).ruleReturnMethods>
@end
<endif>
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{:(<it.type>)<it.name>}; separator=" ">
```

```
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
[<scopeAttributeRefStack()> valueForKey:@"<attr.name>"]
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
[<scopeAttributeRefStack()> setValue:<expr> forKey:@"<attr.name>"];
>>

scopeAttributeRefStack()
 ::= <<
<if(negIndex)>
[<name>_<scope>_stack objectAtIndex:[<name>_<scope>_stack count]-<negIndex>-1]
<else>
<if(index)>
[<name>_<scope>_stack objectAtIndex:<index>]
<else>
[<name>_<scope>_stack lastObject]
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<name>_<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
[<scope> valueForKey:@"<attr.name>"]
<else>
_<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
[_retval setValue: forKey:@"<attr.name>"];
<else>
_<attr.name>
<endif>
>>
```

```
returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
[_retval
 setValue:<expr> forKey:@"<attr.name>"];
<else>
_<attr.name> =<expr>;
<if(LEXER)>
// double check this after beta release!
[_<attr.name> retain];
<endif>
<endif>
>>
/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "_<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "_list_<label>"



// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "[_<scope> text]"
tokenLabelPropertyRef_type(scope,attr) ::= "[_<scope> type]"
tokenLabelPropertyRef_line(scope,attr) ::= "[_<scope> line]"
tokenLabelPropertyRef_pos(scope,attr) ::= "[_<scope> charPositionInLine]"
tokenLabelPropertyRef_channel(scope,attr) ::= "[_<scope> channel]"
tokenLabelPropertyRef_index(scope,attr) ::= "[_<scope> tokenIndex]"
tokenLabelPropertyRef_tree(scope,attr) ::= "_<scope>_tree"

ruleLabelPropertyRef_start(scope,attr)
 ::= "[_<scope> start]"
ruleLabelPropertyRef_stop(scope,attr) ::= "[_<scope> stop]"
ruleLabelPropertyRef_tree(scope,attr) ::= "[_<scope> tree]"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
//[input textForNode:[_<scope> start]]
[input substringWithRange:NSMakeRange([[input treeAdaptor] startIndex:[_<scope> start]], [[input treeAdaptor]
stopIndex:[_<scope> start]])]
<else>
[input substringWithRange:NSMakeRange([_<scope> start], [_<scope> stop]-[_<scope> start])]
<endif>
>>
ruleLabelPropertyRef_st(scope,attr) ::= "[_<scope> st]"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "_<label>"
```

```
lexerRuleLabelPropertyRef_type(scope,attr) ::= "[_<scope> type]"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "[_<scope> line]"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "[_<scope> charPositionInLine]"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "[_<scope> channel]"
lexerRuleLabelPropertyRef_index(scope,attr) ::= "[_<scope> tokenIndex]"
lexerRuleLabelPropertyRef_text(scope,attr)
 ::= "[_<scope> text]"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "[_retval start]"
rulePropertyRef_stop(scope,attr) ::= "[_retval stop]"
rulePropertyRef_tree(scope,attr) ::= "[_retval tree]"
rulePropertyRef_text(scope,attr) ::= "[input substringWithRange:NSMakeRange(_start, [input index]-_start)]"
rulePropertyRef_st(scope,attr) ::= "[_retval st]"

ruleSetPropertyRef_tree(scope,attr,expr) ::= "[_retval setValue:<expr> forKey:@\"tree\"]"
ruleSetPropertyRef_st(scope,attr,expr) ::= "<\n>#error StringTemplates are unsupported<\n>"

/* hideous: find a way to cut down on the number of templates to support read/write access */
/* TODO: also, which ones are valid to write to? ask Ter */
lexerRuleSetPropertyRef_text(scope,attr,expr) ::= "[_state setText:<expr>];"
lexerRuleSetPropertyRef_type(scope,attr,expr) ::= "_type"
lexerRuleSetPropertyRef_line(scope,attr,expr) ::= "[_state tokenStartLine]"
lexerRuleSetPropertyRef_pos(scope,attr,expr)
 ::= "[_state tokenCharPositionInLine]"
lexerRuleSetPropertyRef_index(scope,attr,expr) ::= "-1" // undefined token index in lexer
lexerRuleSetPropertyRef_channel(scope,attr,expr) ::= "[_state setChannel:<expr>];"
lexerRuleSetPropertyRef_start(scope,attr,expr) ::= "[_state tokenStartCharIndex]"
lexerRuleSetPropertyRef_stop(scope,attr,expr) ::= "([self charIndex]-1)"



lexerRulePropertyRef_text(scope,attr) ::= "[self text]"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "[_state tokenStartLine]"
lexerRulePropertyRef_pos(scope,attr) ::= "[_state tokenCharPositionInLine]"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "[_state channel]"
lexerRulePropertyRef_start(scope,attr) ::= "[_state tokenStartCharIndex]"
lexerRulePropertyRef_stop(scope,attr) ::= "([self charIndex]-1)"

/** How to execute an action */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if
 ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
if ( ![_state isBacktracking] ) {
 <action>
}
<endif>
<else>
<action>
<endif>
>>



// M I S C (properties, etc...)

bitset(name, words64) ::= <<
const static unsigned long long <name>_data[] = {<words64:{<it>LL};separator=",">};
static ANTLRBitSet *<name>;
<! ANTLRBitSet *<name> = [[ANTLRBitSet alloc] initWithBits:<name>_data count:<length(words64)>];<\n> !>
>>

bitsetInit(name, words64) ::= <<
<name> = [[ANTLRBitSet alloc] initWithBits:<name>_data count:<length(words64)>];<\n>
>>

codeFileExtension() ::= ".m"
headerFileExtension() ::= ".h"

true() ::= "YES"
false() ::= "NO"
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ObjC/ObjC.stg

No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType>
 _first_0 = null;
<ASTLabelType> _last = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
   retval.tree = (<ASTLabelType>)adaptor.getParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
```

```
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*  setting of _last
*/
```

```
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
>>

/**
 ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate>
 ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
>>
```

```
/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupTree(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

// SET
 AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
```

```
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel>
 = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.getTree());
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(),
 root_<treeLevel>);
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/** Streams for token refs are tree nodes now;
 override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
```

```
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
/* in sync with Java/Java.stg revision 107 */

group Python implements ANTLRCore;

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
```

```
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
          docComment, recognizer,
          name, tokens, tokenNames, rules, cyclicDFAs,
          bitsets, buildTemplate, buildAST, rewriteMode, profile,
          backtracking, synpreds, memoize, numRules,
          fileName,
 ANTLRVersion, generatedTimestamp, trace,
          scopes, superClass, literals) ::=
<<
# $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

<@imports>
import sys
from antlr3 import *
<if(TREE_PARSER)>
from antlr3.tree import *<\n>
<endif>
from antlr3.compat import set, frozenset
<@end>

<actions.(actionScope).header>

<! <docComment> !>

# for convenience in actions
HIDDEN = BaseRecognizer.HIDDEN

# token types
<tokens:{<it.name>=<it.type>}; separator="\n">

<recognizer>

<if(actions.(actionScope).main)>
<actions.(actionScope).main>
<else>
def main(argv, stdin=sys.stdin, stdout=sys.stdout, stderr=sys.stderr):
<if(LEXER)>
   from antlr3.main import LexerMain
   main = LexerMain(<recognizer.name>)<\n>
<endif>
<if(PARSER)>
   from antlr3.main import ParserMain
   main = ParserMain("<recognizer.grammar.name>Lexer", <recognizer.name>)<\n>
<endif>
<if(TREE_PARSER)>
   from antlr3.main import WalkerMain
   main = WalkerMain(<recognizer.name>)<\n>
<endif>
```

```
   main.stdin = stdin
    main.stdout = stdout
   main.stderr = stderr
   main.execute(argv)<\n>
<endif>

<actions.(actionScope).footer>

if __name__ == '__main__':
   main(sys.argv)

>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="Lexer") ::= <<
<grammar.directDelegates:
{g|from <g.recognizerName> import <g.recognizerName>}; separator="\n">

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
   <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>

   grammarFileName = "<fileName>"
   antlr_version = version_str_to_tuple("<ANTLRVersion>")
   antlr_version_str = "<ANTLRVersion>"

   def __init__(self<grammar.delegators:{g|, <g:delegateName()>}>, input=None, state=None):
       if state is None:
           state = RecognizerSharedState()
       super(<grammar.recognizerName>, self).__init__(input, state)

<if(memoize)>
<if(grammar.grammarIsRoot)>
       self._state.ruleMemo = {}
<endif>
<endif>

        <grammar.directDelegates:
        {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)}; separator="\n">
       <grammar.delegators:
        {g|self.<g:delegateName()> = <g:delegateName()>}; separator="\n">
       <last(grammar.delegators):
     {g|self.gParent = <g:delegateName()>}; separator="\n">

       <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">

       <actions.lexer.init>
   <actions.lexer.members>



<if(filterMode)>
   <filteringNextToken()>
<endif>
   <rules; separator="\n\n">

   <synpreds:{p | <lexerSynpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>



>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*
  at backtracking==1.
*/
filteringNextToken() ::= <<
def nextToken(self):
   while True:
       if self.input.LA(1) == EOF:
           return EOF_TOKEN

       self._state.token = None
       self._state.channel = DEFAULT_CHANNEL
       self._state.tokenStartCharIndex = self.input.index()
       self._state.tokenStartCharPositionInLine = self.input.charPositionInLine
       self._state.tokenStartLine = self.input.line
       self._state._text = None
       try:
           m = self.input.mark()
           try:
               # means we won't throw slow exception
               self._state.backtracking = 1
               try:
                   self.mTokens()
               finally:
                   self._state.backtracking = 0

           except BacktrackingFailed:
               # mTokens backtracks with synpred at backtracking==2
               # and we set the synpredgate to allow actions at level 1.
```

```
              self.input.rewind(m)
              self.input.consume()
 # advance one char and try again

          else:
              self.emit()
              return self._state.token

      except RecognitionException, re:
          # shouldn't happen in backtracking mode, but...
          self.reportError(re)
          self.recover(re)



def memoize(self, input, ruleIndex, ruleStartIndex, success):
   if self._state.backtracking > 1:
       # is Lexer always superclass?
       <@superClassName><superClass><@end>.memoize(self, input, ruleIndex, ruleStartIndex, success)



def alreadyParsedRule(self, input, ruleIndex):
   if self._state.backtracking > 1:
       return <@superClassName><superClass><@end>.alreadyParsedRule(self, input, ruleIndex)
   return False



>>

actionGate() ::= "self._state.backtracking == 0"

filteringActionGate() ::= "self._state.backtracking == 1"

/** How to generate a parser */

genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
 filterMode,
            ASTLabelType="Object", labelType, members, rewriteElementType,
            init) ::= <<
<if(grammar.grammarIsRoot)>
# token names
tokenNames = [
   "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>",
   <tokenNames; wrap, separator=", ">
]<\n>
<else>
from <grammar.composite.rootGrammar.recognizerName> import tokenNames<\n>
<endif>
```

```
<scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeClass(scope=it)><endif>}>

<grammar.directDelegates:
{g|from <g.recognizerName> import <g.recognizerName>}; separator="\n">

<rules:{<ruleAttributeScopeClass(scope=it.ruleDescriptor.ruleScope)>}>

class <grammar.recognizerName>(<@superClassName><superClass><@end>):
  grammarFileName = "<fileName>"
  antlr_version = version_str_to_tuple("<ANTLRVersion>")
  antlr_version_str = "<ANTLRVersion>"
  tokenNames = tokenNames

  def __init__(self<grammar.delegators:{g|, <g:delegateName()>}>, input, state=None, *args, **kwargs):
     if state is None:
        state
 = RecognizerSharedState()

     <@args()>
     super(<grammar.recognizerName>, self).__init__(input, state, *args, **kwargs)

<if(memoize)>
<if(grammar.grammarIsRoot)>
     self._state.ruleMemo = {}
<endif>
<endif>

     <cyclicDFAs:{dfa | <cyclicDFAInit(dfa)>}; separator="\n">

     <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScopeStack(scope=it)><endif>}>
 <rules:{<ruleAttributeScopeStack(scope=it.ruleDescriptor.ruleScope)>}>

     <init>

     <grammar.delegators:
      {g|self.<g:delegateName()> = <g:delegateName()>}; separator="\n">
     <grammar.directDelegates:
      {g|self.<g:delegateName()> = <g.recognizerName>(<trunc(g.delegators):{p|<p:delegateName()>, }>self, input,
state)}; separator="\n">
     <!grammar.directDelegates:
      {g|self.<g:delegateName()> = <g.recognizerName>(self<grammar.delegators:{g|, <g:delegateName()>}>,
input, state)}; separator="\n"!>
     <last(grammar.delegators):
    {g|self.gParent = self.<g:delegateName()>};
 separator="\n">

     <@init>
     <@end>
```

```
   <@members>
   <@end>

   <members>

   <rules; separator="\n\n">

   <! generate rule/method definitions for imported rules so they
      appear to be defined in this recognizer. !>
   # Delegated rules
   <grammar.delegatedRules:{ruleDescriptor| <delegateRule(ruleDescriptor)> }; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:{FOLLOW_<it.name>_in_<it.inName><it.tokenIndex> = frozenset([<it.tokenTypes:{<it>};separator=",
">])<\n>}>

>>

delegateRule(ruleDescriptor) ::= <<
def <ruleDescriptor.name>(self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<\ >   <if(ruleDescriptor.hasReturnValue)>return
<endif>self.<ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.att
ributes:{a|<a.name>}; separator=", ">)



>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
 ASTLabelType="object", superClass="Parser", labelType="Token", members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="TokenStream", rewriteElementType="Token", init={<actions.parser.init>}, ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules, numRules, bitsets,
labelType={<ASTLabelType>}, ASTLabelType="Object", superClass="TreeParser",
members={<actions.treeparser.members>}, filterMode) ::= <<
<genericParser(inputStreamType="TreeNodeStream", rewriteElementType="Node",
init={<actions.treeparser.init>}, ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
```

```
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined
 into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
# $ANTLR start "<ruleName>"
def <ruleName>_fragment(self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<if(trace)>
   self.traceIn("<ruleName>_fragment", <ruleDescriptor.index>)
   try:
       <block>

   finally:
       self.traceOut("<ruleName>_fragment", <ruleDescriptor.index>)

<else>
   <block>
<endif>
# $ANTLR end "<ruleName>"


>>

synpred(name) ::= <<
def <name>(self):
   self._state.backtracking += 1
   <@start()>
   start = self.input.mark()
   try:
       self.<name>_fragment()
   except BacktrackingFailed:
       success = False
   else:
       success = True
   self.input.rewind(start)
   <@stop()>
   self._state.backtracking -= 1
   return success


>>

lexerSynpred(name) ::= <<
<synpred(name)>
```

```
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if self._state.backtracking > 0 and self.alreadyParsedRule(self.input,
 <ruleDescriptor.index>):
   # for cached failed rules, alreadyParsedRule will raise an exception
   success = True
   return <ruleReturnValue()>

<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>
if self._state.backtracking > 0:
   raise BacktrackingFailed

<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<returnScope(scope=ruleDescriptor.returnScope)>

# $ANTLR start "<ruleName>"
# <fileName>:<description>
<ruleDescriptor.actions.decorate>
def <ruleName>(self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<if(trace)>
   self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
    <@preamble()>
   <@body><ruleBody()><@end>
   <@postamble()>
   return <ruleReturnValue()>

# $ANTLR end "<ruleName>"
>>

ruleBody() ::= <<
```

```
<if(memoize)>
<if(backtracking)>
success = False<\n>
<endif>
<endif>
try:
  try:
      <ruleMemoization(name=ruleName)>
      <block>
      <ruleCleanUp()>
      <(ruleDescriptor.actions.after):execAction()>

<if(memoize)>
<if(backtracking)>
      success = True<\n>
<endif>
<endif>
<if(exceptions)>
  <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
  <actions.(actionScope).rulecatch>
<else>
  except RecognitionException, re:
      self.reportError(re)
      self.recover(self.input, re)
      <@setErrorReturnValue()>

<endif>
<else>
  finally:
      pass

<endif>
<endif>
finally:
<if(trace)>
  self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
  pass
>>

catch(decl,action)
```

```
 ::= <<
except <e.decl>:
   <e.action>

>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval = self.<ruleDescriptor.name>_return()
retval.start = self.input.LT(1)<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.name> = <if(a.initValue)><a.initValue><else>None<endif>
}>
<endif>
<if(memoize)>
<ruleDescriptor.name>_StartIndex = self.input.index()
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{self.<it>_stack.append(<it>_scope())}; separator="\n">
<ruleDescriptor.ruleScope:{self.<it.name>_stack.append(<it.name>_scope())}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{self.<it>_stack.pop()}; separator="\n">
<ruleDescriptor.ruleScope:{self.<it.name>_stack.pop()}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
   :{<it.label.text> = None}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,
  ruleDescriptor.wildcardTreeListLabels]
   :{list_<it.label.text> = None}; separator="\n"
>
<[ruleDescriptor.ruleLabels,ruleDescriptor.ruleListLabels]
   :ruleLabelDef(label=it); separator="\n"
>
<ruleDescriptor.ruleListLabels:{<it.label.text> = None}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<it.label.text> = None}; separator="\n"
>
<ruleDescriptor.charLabels:{<it.label.text> = None}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{list_<it.label.text> = None}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = self.input.LT(-1)<\n>
<endif>
<endif>
>>

memoize()
 ::= <<
<if(memoize)>
<if(backtracking)>
if self._state.backtracking > 0:
   self.memoize(self.input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex, success)

<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
# $ANTLR start "<ruleName>"
def m<ruleName>(self, <ruleDescriptor.parameterScope:parameterScope(scope=it)>):
<if(trace)>
```

```
    self.traceIn("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
    <ruleScopeSetUp()>
    <ruleDeclarations()>
<if(memoize)>
<if(backtracking)>
    success = False<\n>
<endif>
<endif>
    try:
<if(nakedBlock)>
        <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
        <ruleDescriptor.actions.init>
        <block><\n>
<else>
        _type = <ruleName>
        _channel = DEFAULT_CHANNEL

        <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
         <ruleDescriptor.actions.init>
        <block>
        <ruleCleanUp()>
        self._state.type = _type
        self._state.channel = _channel
        <(ruleDescriptor.actions.after):execAction()>
<endif>
<if(memoize)>
<if(backtracking)>
        success = True<\n>
<endif>
<endif>

    finally:
<if(trace)>
        self.traceOut("<ruleName>", <ruleDescriptor.index>)<\n>
<endif>
 <ruleScopeCleanUp()>
        <memoize()>
        pass

# $ANTLR end "<ruleName>"


>>

/** How to generate code for the implicitly-defined lexer grammar rule
```

```
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
def mTokens(self):
   <block><\n>



>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@body><blockBody()><@end>
>>

blockBody() ::= <<
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<@prebranch()>
<alts:altSwitchCase();
 separator="\nel">
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
# <fileName>:<description>
alt<decisionNumber> = <maxAlt>
<decls>
<@predecision()>
<@decision><decision><@end>
<@postdecision()>
<alts:altSwitchCase(); separator="\nel">
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>
```

```
/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
# <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n)
 ::= <<
# <fileName>:<description>
cnt<decisionNumber> = 0
<decls>
<@preloop()>
<@loopBody>
<positiveClosureBlockLoop()>
<@end>
<@postloop()>
>>

positiveClosureBlockLoop() ::= <<
while True: #loop<decisionNumber>
   alt<decisionNumber> = <maxAlt>
   <@predecision()>
   <@decisionBody><decision><@end>
   <@postdecision()>
   <alts:altSwitchCase(); separator="\nel">
   else:
       if cnt<decisionNumber> >= 1:
           break #loop<decisionNumber>

       <ruleBacktrackFailure()>
       eee = EarlyExitException(<decisionNumber>, self.input)
       <@earlyExitException()>
       raise eee

   cnt<decisionNumber> += 1
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
# <fileName>:<description>
```

```
<decls>
<@preloop()>
<@loopBody>
<closureBlockLoop()>
<@end>
<@postloop()>
>>

closureBlockLoop()
 ::= <<
while True: #loop<decisionNumber>
   alt<decisionNumber> = <maxAlt>
   <@predecision()>
   <@decisionBody><decision><@end>
   <@postdecision()>
   <alts:altSwitchCase(); separator="\nel">
   else:
       break #loop<decisionNumber>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
if alt<decisionNumber> == <i>:
   <@prealt()>
   <it>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt, treeLevel,rew) ::= <<
# <fileName>:<description>
pass <! so empty alternatives are a valid block
 !>
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
```

```
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label>=<endif>self.match(self.input, <token>,
self.FOLLOW_<token>_in_<ruleName><elementIndex>)
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label, elem) ::= <<
if list_<label> is None:
   list_<label> = []
list_<label>.append(<elem>)<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.match(<char>)
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchRange(<a>,
 <b>)
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
if <s>:
   self.input.consume()
   <postmatchCode>
<if(!LEXER)>
   self._state.errorRecovery = False<\n>
<endif>

else:
   <ruleBacktrackFailure()>
   mse = MismatchedSetException(None, self.input)
   <@mismatchedSetException()>
<if(LEXER)>
   self.recover(mse)
   raise mse
<else>
   raise mse
   <! use following code to make it recover inline; remove throw mse;
   self.recoverFromMismatchedSet(
       self.input, mse, self.FOLLOW_set_in_<ruleName><elementIndex>
       )
   !>
<endif>
<\n>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
<label>Start = self.getCharIndex()
self.match(<string>)
<label> = CommonToken(input=self.input,
 type=INVALID_TOKEN_TYPE, channel=DEFAULT_CHANNEL, start=<label>Start, stop=self.getCharIndex()-
1)
```

```
<else>
self.match(<string>)
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label> = self.input.LT(1)<\n>
<endif>
self.matchAny(self.input)
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = self.input.LA(1)<\n>
<endif>
self.matchAny()
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values. The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
self._state.following.append(self.FOLLOW_<rule.name>_in_<ruleName><elementIndex>)
<if(label)><label>
 = <endif>self.<if(scope)><scope:delegateName()>.<endif><rule.name>(<args; separator=", ">)<\n>
self._state.following.pop()
>>

/** ids+=rule */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>
```

```
/** A lexer rule reference
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
self.<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<label> = CommonToken(
   input=self.input,
   type=INVALID_TOKEN_TYPE,
   channel=DEFAULT_CHANNEL,
   start=<label>Start<elementIndex>,
   stop=self.getCharIndex()-1
   )
<else>
self.<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">)
<endif>
>>

/**
 i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
<label>Start<elementIndex> = self.getCharIndex()
self.match(EOF)
<label> = CommonToken(input=self.input, type=EOF, channel=DEFAULT_CHANNEL,
start=<label>Start<elementIndex>, stop=self.getCharIndex()-1)
<else>
self.match(EOF)
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if self.input.LA(1) == DOWN:
   self.match(self.input, DOWN, None)
   <children:element()>
   self.match(self.input, UP, None)

<else>
self.match(self.input, DOWN, None)
<children:element()>
self.match(self.input, UP, None)
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into
 a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if not (<evalPredicate(...)>):
   <ruleBacktrackFailure()>
   raise FailedPredicateException(self.input, "<ruleName>", "<description>")

>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel">
else:
<if(eotPredictsAlt)>
   alt<decisionNumber> = <eotPredictsAlt>
<else>
   <ruleBacktrackFailure()>
   nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
   <@noViableAltException()>
   raise nvae<\n>
<endif>
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState)
 ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel">
>>
```

```
/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = self.input.LA(<k>)<\n>
<edges; separator="\nel"><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber> = <eotPredictsAlt> <! if no edges, don't gen ELSE !>
<else>
else:
   alt<decisionNumber> = <eotPredictsAlt>
<\n>
<endif>
<endif>
>>

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber> = <alt>"

/** A simple edge with an expression.  If the
 expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if (<labelExpr>) <if(predicates)>and (<predicates>)<endif>:
   <targetState>
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
<!
 FIXME: this is one of the few occasion, where I miss a switch statement
 in Python. ATM this is implemented as a list of if .. elif ..
 This may be replaced by faster a dictionary lookup, when I find a solution
 for the cases when an edge is not a plain dfaAcceptState.
!>
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
```

```
else:
<if(eotPredictsAlt)>
  alt<decisionNumber> = <eotPredictsAlt>
<else>
  <ruleBacktrackFailure()>
   nvae = NoViableAltException("<description>", <decisionNumber>, <stateNumber>, self.input)<\n>
  <@noViableAltException()>
  raise nvae<\n>
<endif>

>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
LA<decisionNumber> = self.input.LA(<k>)
<edges; separator="\nel">
<if(eotPredictsAlt)>
else:
  alt<decisionNumber> = <eotPredictsAlt>
<endif>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
if <labels:{LA<decisionNumber> == <it>}; separator=" or ">:
  <targetState>
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = self.dfa<decisionNumber>.predict(self.input)
>>

/*
 Dump DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
# lookup tables for DFA #<dfa.decisionNumber>

DFA<dfa.decisionNumber>_eot = DFA.unpack(
   u"<dfa.javaCompressedEOT; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_eof = DFA.unpack(
   u"<dfa.javaCompressedEOF; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_min = DFA.unpack(
   u"<dfa.javaCompressedMin; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_max = DFA.unpack(
   u"<dfa.javaCompressedMax; wrap="\"\n    u\"">"
   )

DFA<dfa.decisionNumber>_accept = DFA.unpack(
   u"<dfa.javaCompressedAccept; wrap="\"\n
   u\"">"
   )

DFA<dfa.decisionNumber>_special = DFA.unpack(
   u"<dfa.javaCompressedSpecial; wrap="\"\n    u\"">"
   )


DFA<dfa.decisionNumber>_transition = [
   <dfa.javaCompressedTransition:{s|DFA.unpack(u"<s; wrap="\"\nu\"">")}; separator=",\n">
]

# class definition for DFA #<dfa.decisionNumber>

class DFA<dfa.decisionNumber>(DFA):
   pass

   <@errorMethod()>

<if(dfa.specialStateSTs)>
   def specialStateTransition(self_, s, input):
       # convince pylint that my self_ magic is ok ;)
       # pylint: disable-msg=E0213

       # pretend we are a member of the recognizer
```

```
    # thus semantic predicates can be evaluated
    self = self_.recognizer

    _s = s

    <dfa.specialStateSTs:{state |
if s == <i0>: <! compressed special state numbers 0..n-1 !>
  <state>}; separator="\nel">

<if(backtracking)>
    if self._state.backtracking >0:
      raise BacktrackingFailed

<endif>
    nvae = NoViableAltException(self_.getDescription(),
 <dfa.decisionNumber>, _s, input)
    self_.error(nvae)
    raise nvae<\n>
<endif>

>>

cyclicDFAInit(dfa) ::= <<
self.dfa<dfa.decisionNumber> = self.DFA<dfa.decisionNumber>(
  self, <dfa.decisionNumber>,
  eot = self.DFA<dfa.decisionNumber>_eot,
  eof = self.DFA<dfa.decisionNumber>_eof,
  min = self.DFA<dfa.decisionNumber>_min,
  max = self.DFA<dfa.decisionNumber>_max,
  accept = self.DFA<dfa.decisionNumber>_accept,
  special = self.DFA<dfa.decisionNumber>_special,
  transition = self.DFA<dfa.decisionNumber>_transition
  )<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
LA<decisionNumber>_<stateNumber> = input.LA(1)<\n>
<if(semPredState)> <! get next lookahead symbol to test edges, then rewind !>
index<decisionNumber>_<stateNumber> = input.index()
input.rewind()<\n>
<endif>
s = -1
<edges; separator="\nel">
<if(semPredState)>
```

```
 <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>)<\n>
<endif>
if s >= 0:
   return s
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if (<labelExpr>)<if(predicates)> and (<predicates>)<endif>:
   s = <targetStateNumber><\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
se:
   s = <targetStateNumber><\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "((<left>) and (<right>))"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o |  or <o>}>)"

notPredicate(pred) ::= "not (<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "self.<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>
 == <atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "self.input.LA(<k>) == <atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(<lower> \<= LA<decisionNumber>_<stateNumber> \<= <upper>)
>>
```

```
isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(<lower> \<= self.input.LA(<k>) \<= <upper>)"

setTest(ranges) ::= "<ranges; separator=\" or \">"

// A T T R I B U T E S

globalAttributeScopeClass(scope) ::= <<
<if(scope.attributes)>
class <scope.name>_scope(object):
   def __init__(self):
       <scope.attributes:{self.<it.decl> = None}; separator="\n">

<endif>
>>

globalAttributeScopeStack(scope) ::= <<
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
>>

ruleAttributeScopeClass(scope) ::= <<
<if(scope.attributes)>
class <scope.name>_scope(object):

   def __init__(self):
       <scope.attributes:{self.<it.decl> = None}; separator="\n">

<endif>
>>

ruleAttributeScopeStack(scope) ::= <<
<if(scope.attributes)>
self.<scope.name>_stack = []<\n>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<label.label.text> = None<\n>
>>

returnStructName() ::= "<it.name>_return"
```

```
/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
class <ruleDescriptor:returnStructName()>(<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope):
   def __init__(self):
       super(<grammar.recognizerName>.<ruleDescriptor:returnStructName()>, self).__init__()

       <scope.attributes:{self.<it.decl> = None}; separator="\n">
       <@ruleReturnInit()>


    <@ruleReturnMembers()>

<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> = <expr>"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
self.<scope>_stack[-<negIndex>].<attr.name>
<else>
<if(index)>
self.<scope>_stack[<index>].<attr.name>
<else>
self.<scope>_stack[-1].<attr.name>
<endif>
<endif>
>>

/* not applying patch because of bug in action parser!

<if(negIndex)>
((len(self.<scope>_stack) - <negIndex> - 1) >= 0 and [self.<scope>_stack[-<negIndex>].<attr.name>] or [None])[0]
<else>
<if(index)>
((<index> \< len(self.<scope>_stack)) and [self.<scope>_stack[<index>].<attr.name>] or [None])[0]
<else>
((len(self.<scope>_stack) > 0) and [self.<scope>_stack[-1].<attr.name>] or [None])[0]
```

```
<endif>
<endif>

*/

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
<!FIXME: this seems not to be used by ActionTranslator...!>
self.<scope>_stack[-<negIndex>].<attr.name>
 = <expr>
<else>
<if(index)>
<!FIXME: this seems not to be used by ActionTranslator...!>
self.<scope>_stack[<index>].<attr.name> = <expr>
<else>
self.<scope>_stack[-1].<attr.name> = <expr>
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "self.<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
((<scope> is not None) and [<scope>.<attr.name>] or [None])[0]
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
 = <expr>
<else>
```

```
<attr.name> = <expr>
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr) ::= "<scope>.text"
tokenLabelPropertyRef_type(scope,attr) ::= "<scope>.type"
tokenLabelPropertyRef_line(scope,attr) ::= "<scope>.line"
tokenLabelPropertyRef_pos(scope,attr) ::= "<scope>.charPositionInLine"
tokenLabelPropertyRef_channel(scope,attr) ::= "<scope>.channel"
tokenLabelPropertyRef_index(scope,attr) ::= "<scope>.index"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"

ruleLabelPropertyRef_start(scope,attr) ::= "<scope>.start"
ruleLabelPropertyRef_stop(scope,attr) ::= "<scope>.stop"
ruleLabelPropertyRef_tree(scope,attr) ::= "<scope>.tree"
ruleLabelPropertyRef_text(scope,attr)
 ::= <<
<if(TREE_PARSER)>
((<scope> is not None) and [self.input.getTokenStream().toString(
   self.input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
   self.input.getTreeAdaptor().getTokenStopIndex(<scope>.start)
   )] or [None])[0]
<else>
((<scope> is not None) and [self.input.toString(<scope>.start,<scope>.stop)] or [None])[0]
<endif>
>>
ruleLabelPropertyRef_st(scope,attr) ::= "((<scope> is not None) and [<scope>.st] or [None])[0]"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::= "((<scope> is not None) and [<scope>.type] or [0])[0]"
lexerRuleLabelPropertyRef_line(scope,attr) ::= "((<scope> is not None) and [<scope>.line] or [0])[0]"
lexerRuleLabelPropertyRef_pos(scope,attr) ::= "((<scope> is not None) and [<scope>.charPositionInLine] or
[0])[0]"
lexerRuleLabelPropertyRef_channel(scope,attr) ::= "((<scope> is not None) and [<scope>.channel] or [0])[0]"
lexerRuleLabelPropertyRef_index(scope,attr)
```

```
 ::= "((<scope> is not None) and [<scope>.index] or [0])[0]"
lexerRuleLabelPropertyRef_text(scope,attr) ::= "((<scope> is not None) and [<scope>.text] or [None])[0]"
lexerRuleLabelPropertyRef_int(scope,attr) ::= "((<scope> is not None) and [int(<scope>.text)] or [0])[0]"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "retval.start"
rulePropertyRef_stop(scope,attr) ::= "retval.stop" //mmm... or input.LT(-1)??
rulePropertyRef_tree(scope,attr) ::= "retval.tree"
rulePropertyRef_text(scope,attr) ::= "self.input.toString(retval.start, self.input.LT(-1))"
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "self.text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "self._state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "self._state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::=
 "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "self._state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(self.getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int(<scope>.text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>"


/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
<if(actions.(actionScope).synpredgate)>
if <actions.(actionScope).synpredgate>:
   <action>

<else>
if <actions.(actionScope).synpredgate>:
   <action>

<endif>
<else>
#action start
<action>
#action end
<endif>
>>

/** How to always execute an action even when backtracking */
```

```
execForcedAction(action) ::= "<action>"


// M I S C (properties, etc...)

codeFileExtension()
 ::= ".py"

true() ::= "True"
false() ::= "False"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/Python.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/ASTDbg.stg
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/Transition.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Python/AST.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/OrderedHashSet.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/SemanticContext.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/tool/NFAFactory.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/DecisionProbe.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/CPPTarget.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/StateCluster.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/ActionScriptTarget.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/Utils.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-

jar/org/antlr/codegen/templates/ActionScript/ASTParser.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/ASTParser.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/ACyclicDFACodeGenerator.java
*
 /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/Dbg.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/IntArrayList.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/Interval.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/JavaTarget.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/ST.stg
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NFA.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/DFA.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/Target.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/IntSet.java
*
 /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NFAConfiguration.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/LookaheadSet.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/DFAOptimizer.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/IntervalSet.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/DFAState.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/Label.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/RuleClosureTransition.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NFAContext.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/State.java
*
 /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/misc/BitSet.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NFAState.java
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/analysis/NFAToDFAConverter.java
No license file was found, but licenses were detected in source scan.

notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
group AST;

@outputFile.imports() ::= <<
<@super.imports()>
<if(!TREE_PARSER)><! tree parser would already have imported !>
import org.antlr.runtime.tree.*;<\n>
<endif>
>>

@genericParser.members() ::= <<
<@super.members()>
<parserMembers()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
protected TreeAdaptor adaptor = new CommonTreeAdaptor();<\n>
public void setTreeAdaptor(TreeAdaptor adaptor) {
   this.adaptor = adaptor;
   <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
}
public
 TreeAdaptor getTreeAdaptor() {
   return adaptor;
}
>>

@returnScope.ruleReturnMembers() ::= <<
<ASTLabelType> tree;
```

```
public Object getTree() { return tree; }
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
<ASTLabelType> root_0 = null;<\n>
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<[ruleDescriptor.tokenLabels,ruleDescriptor.wildcardTreeLabels,
 ruleDescriptor.wildcardTreeListLabels]:{<ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<ASTLabelType> <it.label.text>_tree=null;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites
   :{RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>");}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites
   :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
 <it>");}; separator="\n">
>>

/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
<if(!rewriteMode)>
root_0 = (<ASTLabelType>)adaptor.nil();<\n>
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** ids+=ID and track it for use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
```

```
tokenRefTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track
 for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<token>.add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule.name>.add(<label>.getTree());
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>stream_<rule>.add(<label>.getTree());
>>

/**
 ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".getTree()",...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
 referencedWildcardLabels,
 referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements: <referencedElementsDeep; separator=", ">
// token labels: <referencedTokenLabels; separator=", ">
// rule labels: <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels: <referencedRuleListLabels; separator=", ">
// wildcard labels: <[referencedWildcardLabels,referencedWildcardListLabels]; separator=", ">
<if(backtracking)>
if ( <actions.(actionScope).synpredgate>
 ) {<\n>
<endif>
<prevRuleRootRef()>.tree = root_0;
<rewriteCodeLabels()>
root_0 = (<ASTLabelType>)adaptor.nil();
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
input.replaceChildren(adaptor.getParent(retval.start),
                      adaptor.getChildIndex(retval.start),
                      adaptor.getChildIndex(_last),
                      retval.tree);
<endif>
<endif>
<! if parser or tree-parser && rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.tree = root_0;
<else>
<if(!rewriteMode)>
<prevRuleRootRef()>.tree = root_0;
<endif>
<endif>
<if(backtracking)>
}
<endif>
>>

rewriteCodeLabels() ::= <<
```

```
<referencedTokenLabels
  :{RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>",<it>);};
  separator="\n"
>
<referencedTokenListLabels
   :{RewriteRule<rewriteElementType>Stream stream_<it>=new
RewriteRule<rewriteElementType>Stream(adaptor,"token <it>", list_<it>);};
  separator="\n"
>
<referencedWildcardLabels
  :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",<it>);};
  separator="\n"
>
<referencedWildcardListLabels
  :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"wildcard <it>",list_<it>);};
  separator="\n"
>
<referencedRuleLabels
  :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"rule
<it>",<it>!=null?<it>.tree:null);};
  separator="\n"
>
<referencedRuleListLabels
  :{RewriteRuleSubtreeStream stream_<it>=new RewriteRuleSubtreeStream(adaptor,"token <it>",list_<it>);};
  separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep,
 // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
if ( <referencedElementsDeep:{el | stream_<el>.hasNext()}; separator="||"> ) {
  <alt>
}
<referencedElementsDeep:{el | stream_<el>.reset();<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
// <fileName>:<description>
while ( <referencedElements:{el | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements, // elements in immediately block; no nested blocks
 description) ::=
<<
if ( !(<referencedElements:{el | stream_<el>.hasNext()}; separator="||">) ) {
   throw new RewriteEarlyExitException();
}
while ( <referencedElements:{el
 | stream_<el>.hasNext()}; separator="||"> ) {
   <alt>
}
<referencedElements:{el | stream_<el>.reset();<\n>}>
>>

rewriteAlt(a) ::= <<
// <a.description>
<if(a.pred)>
if (<a.pred>) {
   <a.alt>
}<\n>
<else>
{
   <a.alt>
}<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "root_0 = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
// <fileName>:<description>
{
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<root:rewriteElement()>
```

```
<children:rewriteElement()>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
}<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
adaptor.addChild(root_<treeLevel>, <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>,
 stream_<label>.nextNode());<\n>
>>

/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextNode());<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($label ...) where label+=... */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<createRewriteNodeFromElement(...)>,
root_<treeLevel>);<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
adaptor.addChild(root_<treeLevel>, <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<createImaginaryNode(tokenType=token,
 ...)>, root_<treeLevel>);<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
root_0 = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of root_0 right
*  before I set it during rewrites.  The assign will be to retval.tree.
*/
prevRuleRootRef() ::= "retval"

rewriteRuleRef(rule) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<rule>.nextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<rule>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteNodeAction(action) ::= <<
adaptor.addChild(root_<treeLevel>, <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<action>, root_<treeLevel>);<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule
 */
rewriteRuleLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
```

```
rewriteRuleListLabelRefRoot(label) ::= <<
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(stream_<label>.nextNode(), root_<treeLevel>);<\n>
>>

rewriteWildcardLabelRef(label) ::= <<
adaptor.addChild(root_<treeLevel>, stream_<label>.nextTree());<\n>
>>


createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel, args) !>
new <hetero>(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
(<ASTLabelType>)adaptor.create(<tokenType>,
 <args; separator=", "><if(!args)>"<tokenType>"<endif>)
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.nextToken()<if(args)>, <args; separator=", "><endif>)
<else>
<if(args)> <! must create new node from old !>
adaptor.create(<token>, <args; separator=", ">)
<else>
stream_<token>.nextNode()
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/AST.stg
No license file was found, but licenses were detected in source scan.

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations()
 ::= <<
<super.ruleDeclarations()>
<ASTLabelType> _first_0 = null;
<ASTLabelType> _last = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(rewriteMode)>
retval.tree = (<ASTLabelType>)_first_0;
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
   retval.tree = (<ASTLabelType>)adaptor.getParent(retval.tree);
<endif>
<if(backtracking)>}<endif>
>>

/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = (<ASTLabelType>)input.LT(1);
{
<ASTLabelType> _save_last_<treeLevel> = _last;
<ASTLabelType> _first_<treeLevel> = null;
<if(!rewriteMode)>
<ASTLabelType> root_<treeLevel> = (<ASTLabelType>)adaptor.nil();
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if
 ( state.backtracking==0 )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input, Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
```

```
*  setting of _last
*/
tokenRefBang(token,label,elementIndex) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
>>

/**
 ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( state.backtracking==0 )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(hetero)>
<label>_tree
 = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
```

```
>>

/** Match ^(label+=TOKEN ...) auto construct */
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

// SET AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode)
 ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.matchSet(...)>
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = (<ASTLabelType>)adaptor.dupNode(<label>);
<endif><\n>
root_<treeLevel> = (<ASTLabelType>)adaptor.becomeRoot(<label>_tree, root_<treeLevel>);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.getTree());
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope)
 ::= <<
<ruleRef(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) <endif>root_<treeLevel> =
(<ASTLabelType>)adaptor.becomeRoot(<label>.getTree(), root_<treeLevel>);
<endif>
>>

/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".getTree()",...)>
>>

/** Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
```

```
new <hetero>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>if ( state.backtracking==0 ) {<\n><endif>
retval.tree = (<ASTLabelType>)adaptor.rulePostProcessing(root_0);
<if(backtracking)>}<endif>
<endif>
>>
```

Found
 in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Perl5/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/

interface ANTLRCore;

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals);

/**
 The header file; make sure to define headerFileExtension() below */
optional
headerFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules, cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals);

lexer(grammar, name, tokens, scopes, rules, numRules, labelType,
     filterMode, superClass);

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
      bitsets, ASTLabelType, superClass,
      labelType, members);

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
           numRules, bitsets, labelType, ASTLabelType,
           superClass, members, filterMode);

/** A simpler version of a rule template that is specific
 to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
```

```
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock);

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize);

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize);

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor);

filteringNextToken();

filteringActionGate();

//
 S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description);

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description);

/** A (..)+ block with 0 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

positiveClosureBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

/** A (..)* block with 0 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

closureBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);
```

```
/**
 Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

optionalBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description);

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew);

// E L E M E N T S

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero);

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero);

listLabel(label,elem);

/** match a character */
charRef(char,label);

/** match a character range */
charRangeRef(a,b,label);

/** For now, sets are interval tests and must be
 tested inline */
matchSet(s,label,elementIndex,postmatchCode);

matchSetAndListLabel(s,label,elementIndex,postmatchCode);

/** Match a string literal */
lexerStringRef(string,label);

wildcard(label,elementIndex);

wildcardAndListLabel(label,elementIndex);

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex);

wildcardCharListLabel(label, elementIndex);

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.
```

```
*/
ruleRef(rule,label,elementIndex,args,scope);

/** ids+=ID */
ruleRefAndListLabel(rule,label,elementIndex,args,scope);

/** A lexer rule reference */
lexerRuleRef(rule,label,args,elementIndex,scope);

/** i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope);

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex);

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel);

/** Every predicate is used as a validating
 predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description);

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState);

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*
*  If a semPredState, don't force lookahead lookup; preds might not
*  need.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState);

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a
*  rule anything other than 'a' predicts exiting.
 *
*  If a semPredState, don't force lookahead lookup; preds might not
*  need.
*/
```

```
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState);

/** An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt);

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates);

// Fixed DFA  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState);

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState);

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState);

dfaEdgeSwitch(labels, targetState);

// Cyclic DFA

/**
 The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description);

/** Generate the tables and support code needed for the DFAState object
*  argument.  Unless there is a semantic predicate (or syn pred, which
*  become sem preds), all states should be encoded in the state tables.
*  Consequently, cyclicDFAState/cyclicDFAEdge,eotDFAEdge templates are
*  not used except for special DFA states that cannot be encoded as
*  a transition table.
*/
cyclicDFA(dfa);

/** A special state in a cyclic DFA; special means has a semantic predicate
*  or it's a huge set of symbols to check.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState);

/** Just like a fixed DFA edge, test the lookahead and indicate what
```

```
*  state to jump to next if successful.  Again, this is for special
*  states.
*/
cyclicDFAEdge(labelExpr,
 targetStateNumber, edgeNumber, predicates);

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates);

// D F A  E X P R E S S I O N S

andPredicates(left,right);

orPredicates(operands);

notPredicate(pred);

evalPredicate(pred,description);

evalSynPredicate(pred,description);

lookaheadTest(atom,k,atomAsInt);

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt);

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt);

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt);

setTest(ranges);

// A T T R I B U T E S

parameterAttributeRef(attr);
parameterSetAttributeRef(attr,expr);

scopeAttributeRef(scope,attr,index,negIndex);
scopeSetAttributeRef(scope,attr,expr,index,negIndex);

/** $x is either global scope
 or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
```

```
*/
isolatedDynamicScopeRef(scope);

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr);

returnAttributeRef(ruleDescriptor,attr);
returnSetAttributeRef(ruleDescriptor,attr,expr);

/** How to translate $tokenLabel */
tokenLabelRef(label);

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label);

// not sure the next are the right approach; and they are evaluated early;
// they cannot see TREE_PARSER or PARSER attributes for example. :(

tokenLabelPropertyRef_text(scope,attr);
tokenLabelPropertyRef_type(scope,attr);
tokenLabelPropertyRef_line(scope,attr);
tokenLabelPropertyRef_pos(scope,attr);
tokenLabelPropertyRef_channel(scope,attr);
tokenLabelPropertyRef_index(scope,attr);
tokenLabelPropertyRef_tree(scope,attr);

ruleLabelPropertyRef_start(scope,attr);
ruleLabelPropertyRef_stop(scope,attr);
ruleLabelPropertyRef_tree(scope,attr);
ruleLabelPropertyRef_text(scope,attr);
ruleLabelPropertyRef_st(scope,attr);

/**
 Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label);

lexerRuleLabelPropertyRef_type(scope,attr);
lexerRuleLabelPropertyRef_line(scope,attr);
lexerRuleLabelPropertyRef_pos(scope,attr);
lexerRuleLabelPropertyRef_channel(scope,attr);
lexerRuleLabelPropertyRef_index(scope,attr);
lexerRuleLabelPropertyRef_text(scope,attr);

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr);
rulePropertyRef_stop(scope,attr);
rulePropertyRef_tree(scope,attr);
rulePropertyRef_text(scope,attr);
```

```
rulePropertyRef_st(scope,attr);

lexerRulePropertyRef_text(scope,attr);
lexerRulePropertyRef_type(scope,attr);
lexerRulePropertyRef_line(scope,attr);
lexerRulePropertyRef_pos(scope,attr);
/** Undefined, but present for consistency with Token attributes; set to -1 */
lexerRulePropertyRef_index(scope,attr);
lexerRulePropertyRef_channel(scope,attr);
lexerRulePropertyRef_start(scope,attr);
lexerRulePropertyRef_stop(scope,attr);

ruleSetPropertyRef_tree(scope,attr,expr);
ruleSetPropertyRef_st(scope,attr,expr);

/**
 How to execute an action */
execAction(action);

// M I S C (properties, etc...)

codeFileExtension();

/** Your language needs a header file; e.g., ".h" */
optional headerFileExtension();

true();
false();
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ANTLRCore.sti
No license file was found, but licenses were detected in source scan.

```
/** Template overrides to add debugging to AST stuff.  Dynamic inheritance
*  hierarchy is set up as ASTDbg : AST : Dbg : Java by code generator.
*/
group ASTDbg;

parserMembers() ::= <<
protected DebugTreeAdaptor adaptor;
public void setTreeAdaptor(TreeAdaptor adaptor) {
<if(grammar.grammarIsRoot)>
   this.adaptor = new DebugTreeAdaptor(dbg,adaptor);
<else>
   this.adaptor = (DebugTreeAdaptor)adaptor; // delegator sends dbg adaptor
<endif><\n>
   <grammar.directDelegates:{g|<g:delegateName()>.setTreeAdaptor(this.adaptor);}>
}
public
 TreeAdaptor getTreeAdaptor() {
   return adaptor;
}<\n>
>>

parserCtorBody() ::= <<
<super.parserCtorBody()>
>>

createListenerAndHandshake() ::= <<
DebugEventSocketProxy proxy =
   new DebugEventSocketProxy(this,port,<if(TREE_PARSER)>input.getTreeAdaptor()<else>adaptor<endif>);
setDebugListener(proxy);
set<inputStreamType>(new Debug<inputStreamType>(input,proxy));
try {
   proxy.handshake();
}
```

```
catch (IOException ioe) {
   reportError(ioe);
}
>>

@ctorForRootGrammar.finally() ::= <<
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
proxy.setTreeAdaptor(adap);
>>

@ctorForProfilingRootGrammar.finally() ::=<<
TreeAdaptor adap = new CommonTreeAdaptor();
setTreeAdaptor(adap);
proxy.setTreeAdaptor(adap);
>>

@ctorForPredefinedListener.superClassRef() ::= "super(input, dbg);"

@ctorForPredefinedListener.finally() ::=<<
<if(grammar.grammarIsRoot)> <! don't create new adaptor for delegates !>
TreeAdaptor adap = new
 CommonTreeAdaptor();
setTreeAdaptor(adap);<\n>
<endif>
>>

@rewriteElement.pregen() ::= "dbg.location(<e.line>,<e.pos>);"
```

Found in path(s):

* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/ASTDbg.stg

No license file was found, but licenses were detected in source scan.

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
/** Templates for building ASTs during tree parsing.
*
*  Deal with many combinations.  Dimensions are:
*  Auto build or rewrite
*    no label, label, list label  (label/no-label handled together)
*    child, root
*    token, set, rule, wildcard
*
*  Each combination has its own template except that label/no label
*  is combined into tokenRef, ruleRef, ...
*/
group ASTTreeParser;

/** Add a variable to track last element matched */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
var
 _first_0:<ASTLabelType> = null;
var _last:<ASTLabelType> = null;<\n>
>>

/** What to emit when there is no rewrite rule.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= <<
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(rewriteMode)>
retval.tree = <ASTLabelType>(_first_0);
if ( adaptor.getParent(retval.tree)!=null && adaptor.isNil( adaptor.getParent(retval.tree) ) )
   retval.tree = <ASTLabelType>(adaptor.getParent(retval.tree));
<endif>
<if(backtracking)>}<endif>
>>
```

```
/** match ^(root children) in tree parser; override here to
*  add tree construction actions.
*/
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
_last = <ASTLabelType>(input.LT(1));
{
var _save_last_<treeLevel>:<ASTLabelType> = _last;
var _first_<treeLevel>:<ASTLabelType> = null;
<if(!rewriteMode)>
var root_<treeLevel>:<ASTLabelType> = <ASTLabelType>(adaptor.nil());
<endif>
<root:element()>
<if(rewriteMode)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> )<endif>
<if(root.el.rule)>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>.tree;
<else>
if ( _first_<enclosingTreeLevel>==null ) _first_<enclosingTreeLevel> = <root.el.label>;
<endif>
<endif>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN ) {
   matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
matchStream(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
matchStream(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
<if(!rewriteMode)>
adaptor.addChild(root_<enclosingTreeLevel>, root_<treeLevel>);
<endif>
_last = _save_last_<treeLevel>;
}<\n>
>>

// TOKEN AST STUFF

/** ID! and output=AST (same as plain tokenRef) 'cept add
*
  setting of _last
*/
```

```
tokenRefBang(token,label,elementIndex) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.tokenRef(...)>
>>

/** ID auto construct */
tokenRef(token,label,elementIndex,hetero) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

/** label+=TOKEN auto construct */
tokenRefAndListLabel(token,label,elementIndex) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) auto construct */
tokenRefRuleRoot(token,label,elementIndex,hetero) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.tokenRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if
 ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>_tree, root_<treeLevel>));
<if(backtracking)>}<endif>
<endif>
>>

/** Match ^(label+=TOKEN ...) auto construct */
```

```
tokenRefRuleRootAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRoot(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard and auto dup the node/subtree */
wildcard(token,label,elementIndex,hetero) ::= <<
_last = (<ASTLabelType>)input.LT(1);
<super.wildcard(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<label>_tree = (<ASTLabelType>)adaptor.dupTree(<label>);
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<else> <! rewrite mode !>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> )<endif>
if
 ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>;
<endif>
>>

// SET AST

matchSet(s,label,hetero,elementIndex,postmatchCode) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree = new <hetero>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
adaptor.addChild(root_<treeLevel>, <label>_tree);
<if(backtracking)>}<endif>
<endif>
}
)>
>>

matchRuleBlockSet(s,label,hetero,elementIndex,postmatchCode,treeLevel="0") ::= <<
<matchSet(...)>
<noRewrite()> <! set return tree !>
>>

matchSetBang(s,label,elementIndex,postmatchCode) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.matchSet(...)>
```

```
>>

matchSetRuleRoot(s,label,hetero,elementIndex,debug) ::= <<
<super.matchSet(..., postmatchCode={
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<endif>
<if(hetero)>
<label>_tree
 = new <hetero>(<label>);
<else>
<label>_tree = <ASTLabelType>(adaptor.dupNode(<label>));
<endif><\n>
root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>_tree, root_<treeLevel>));
<if(backtracking)>}<endif>
<endif>
}
)>
>>

// RULE REF AST

/** rule auto construct */
ruleRef(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRef(...)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) <endif>
<if(!rewriteMode)>
adaptor.addChild(root_<treeLevel>, <label>.tree);
<else> <! rewrite mode !>
if ( _first_<treeLevel>==null ) _first_<treeLevel> = <label>.tree;
<endif>
>>

/** x+=rule auto construct */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label+".tree",...)>
>>

/** ^(rule ...) auto construct */
ruleRefRuleRoot(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRef(...)>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate>
 ) <endif>root_<treeLevel> = <ASTLabelType>(adaptor.becomeRoot(<label>.tree, root_<treeLevel>));
<endif>
>>
```

```
/** ^(x+=rule ...) auto construct */
ruleRefRuleRootAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<listLabel(elem=label+".tree",...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefTrack(...)>
>>

/** x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefTrackAndListLabel(...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefRootTrack(...)>
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
_last = <ASTLabelType>(input.LT(1));
<super.ruleRefRuleRootTrackAndListLabel(...)>
>>

/**
 Streams for token refs are tree nodes now; override to
*  change nextToken to nextNode.
*/
createRewriteNodeFromElement(token,hetero,scope) ::= <<
<if(hetero)>
new <hetero>(stream_<token>.nextNode())
<else>
stream_<token>.nextNode()
<endif>
>>

ruleCleanUp() ::= <<
<super.ruleCleanUp()>
<if(!rewriteMode)>
<if(backtracking)>if ( <actions.(actionScope).synpredgate> ) {<\n><endif>
retval.tree = <ASTLabelType>(adaptor.rulePostProcessing(root_0));
<if(backtracking)>}<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/ActionScript/ASTTreeParser.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2008 Erik van Bilsen
Copyright (c) 2007-2008 Johannes Luber
Copyright (c) 2005-2007 Kunle Odutola
Copyright (c) 2005 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
group AST;

@outputFile.imports() ::= <<
<@super.imports()><if(!TREE_PARSER)><! tree parser would already have imported !>
Antlr.Runtime.Tree,<\n><endif>
>>

@genericParser.members() ::= <<
```

```
<@super.members()>
<parserMembers()>
>>

@genericParser.membersConstructor() ::= <<
<@super.membersConstructor()>
<parserMembersConstructor()>
>>

@genericParser.membersImplementation()
 ::= <<
<@super.membersImplementation()>
<parserMembersImplementation()>
>>

/** Add an adaptor property that knows how to build trees */
parserMembers() ::= <<
 strict protected
   FAdaptor: ITreeAdaptor;
   procedure SetAdaptor(const Value: ITreeAdaptor);
   property Adaptor: ITreeAdaptor read FAdaptor;
 public
   property TreeAdaptor: ITreeAdaptor read FAdaptor write SetAdaptor;

>>

parserMembersConstructor() ::= <<
FAdaptor := TCommonTreeAdaptor.Create;
>>

parserMembersImplementation() ::= <<
procedure T<grammar.recognizerName>.SetAdaptor(const Value: ITreeAdaptor);
begin
 FAdaptor := Value;
 <grammar.directDelegates:{g|<g:delegateName()>.TreeAdaptor := FAdaptor;}>
end;
>>

@returnScope.ruleReturnMembers() ::= <<
function T<grammar.recognizerName>.T<ruleDescriptor:returnStructName()>.GetTree: IANTLRInterface;
begin
 Result := FTree;
end;

procedure T<grammar.recognizerName>.T<ruleDescriptor:returnStructName()>.SetTree(const Value:
IANTLRInterface);
begin
 FTree := Value
 as I<ASTLabelType>;
end;
>>

@returnScopeDeclaration.ruleReturnMembers() ::= <<
strict private
 FTree: I<ASTLabelType>;
protected
 { IRuleReturnScope }
 function GetTree: IANTLRInterface; override;
 procedure SetTree(const Value: IANTLRInterface); override;
>>

/** Add a variable to track rule's return AST */
ruleDeclarations() ::= <<
<super.ruleDeclarations()>
Root[0] := nil;<\n>
>>

ruleDeclarationVars() ::= <<
<super.ruleDeclarationVars()>
Root: array [0..63] of I<ASTLabelType>;
>>

ruleLabelDefs() ::= <<
<super.ruleLabelDefs()>
<ruleDescriptor.tokenLabels:{<it.label.text>_tree := nil;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<it.label.text>_tree := nil;}; separator="\n">
<ruleDescriptor.allTokenRefsInAltsWithRewrites:{Locals['Stream_<it>'] :=
TRewriteRule<rewriteElementType>Stream.Create(Adaptor,'token <it>');}; separator="\n">
<ruleDescriptor.allRuleRefsInAltsWithRewrites:{Locals['Stream_<it>'] :=
TRewriteRuleSubtreeStream.Create(Adaptor,'rule <it>');}; separator="\n">
>>

ruleLabelDefVars()
 ::= <<
<super.ruleLabelDefVars()>
<ruleDescriptor.tokenLabels:{<it.label.text>_tree: I<ASTLabelType>;}; separator="\n">
<ruleDescriptor.tokenListLabels:{<it.label.text>_tree: I<ASTLabelType>;}; separator="\n">
>>
/** When doing auto AST construction, we must define some variables;
*  These should be turned off if doing rewrites.  This must be a "mode"
*  as a rule could have both rewrite and AST within the same alternative
*  block.
*/
@alt.declarations() ::= <<
<if(autoAST)>
<if(outerAlt)>
```

```
<if(!rewriteMode)>
Root[0] := Adaptor.GetNilNode as I<ASTLabelType>;
<endif>
<endif>
<endif>
>>

// Tracking Rule Elements

/** ID and track it for use in a rewrite rule */
tokenRefTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)> <! Track implies no auto AST construction!>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<token>'] as
IRewriteRuleElementStream).Add(<label>);<\n>
>>

/** ids+=ID and track it for
 use in a rewrite rule; adds to ids *and*
*  to the tracking list stream_ID for use in the rewrite.
*/
tokenRefTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefTrack(...)>
<listLabel(elem=label,...)>
>>

/** ^(ID ...) track for rewrite */
tokenRefRuleRootTrack(token,label,elementIndex,hetero) ::= <<
<tokenRefBang(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<token>'] as
IRewriteRuleElementStream).Add(<label>);<\n>
>>

/** Match ^(label+=TOKEN ...) track for rewrite */
tokenRefRuleRootTrackAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRefRuleRootTrack(...)>
<listLabel(elem=label,...)>
>>

wildcardTrack(label,elementIndex) ::= <<
<super.wildcard(...)>
>>

/** rule when output=AST and tracking for rewrite */
ruleRefTrack(rule,label,elementIndex,args,scope) ::= <<
<super.ruleRef(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<rule.name>'] as
IRewriteRuleElementStream).Add(<label>.Tree);<\n>
>>
```

```
/**
 x+=rule when output=AST and tracking for rewrite */
ruleRefTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

/** ^(rule ...) rewrite */
ruleRefRuleRootTrack(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRoot(...)>
<if(backtracking)>if (State.Backtracking = 0) then <endif>(Locals['Stream_<rule>'] as
IRewriteRuleElementStream).Add(<label>.Tree);
>>

/** ^(x+=rule ...) rewrite */
ruleRefRuleRootTrackAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRefRuleRootTrack(...)>
<listLabel(elem=label+".Tree",...)>
>>

// R e w r i t e

rewriteCode(
 alts, description,
 referencedElementsDeep, // ALL referenced elements to right of ->
 referencedTokenLabels,
 referencedTokenListLabels,
 referencedRuleLabels,
 referencedRuleListLabels,
   referencedWildcardLabels,
   referencedWildcardListLabels,
 rewriteBlockLevel, enclosingTreeLevel, treeLevel) ::=
<<

// AST REWRITE
// elements:          <referencedElementsDeep;
 separator=", ">
// token labels:      <referencedTokenLabels; separator=", ">
// rule labels:       <referencedRuleLabels; separator=", ">
// token list labels: <referencedTokenListLabels; separator=", ">
// rule list labels:  <referencedRuleListLabels; separator=", ">
<if(backtracking)>
if (State.Backtracking = 0) then
begin<\n>
<endif>
<prevRuleRootRef()>.Tree := Root[0];
<rewriteCodeLabels()>
```

```
Root[0] := Adaptor.GetNilNode as I<ASTLabelType>;
<alts:rewriteAlt(); separator="else ">
<! if tree parser and rewrite=true !>
<if(TREE_PARSER)>
<if(rewriteMode)>
<prevRuleRootRef()>.Tree = (<ASTLabelType>)adaptor.rulePostProcessing(root[0]);
input.ReplaceChildren(adaptor.GetParent(retval.Start),
                      adaptor.GetChildIndex(retval.Start),
                      adaptor.GetChildIndex(_last),
                      retval.Tree);
<endif>
<endif>
<! if parser or rewrite!=true, we need to set result !>
<if(!TREE_PARSER)>
<prevRuleRootRef()>.Tree := Root[0];<\n>
<endif>
<if(!rewriteMode)>
<prevRuleRootRef()>.Tree
 := Root[0];<\n>
<endif>
<if(backtracking)>
end;
<endif>
>>

rewriteCodeLabels() ::= <<
<referencedTokenLabels
   :{Locals['Stream_<it>'] := TRewriteRule<rewriteElementType>Stream.Create(Adaptor, 'token <it>', <it>);};
   separator="\n"
>
<referencedTokenListLabels
   :{Locals['Stream_<it>'] := TRewriteRule<rewriteElementType>Stream.Create(Adaptor,'token <it>', list_<it>);};
   separator="\n"
>
<referencedRuleLabels:{
if Assigned(<it>) then
 Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', <it>.Tree)
else
 Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', nil);}; separator="\n">
<referencedRuleListLabels
   :{Locals['Stream_<it>'] := TRewriteRuleSubtreeStream.Create(Adaptor, 'token <it>', list_<it>);};
   separator="\n"
>
>>

/** Generate code for an optional rewrite block; note it uses the deep ref'd element
 *  list rather shallow like other blocks.
 */
```

```
rewriteOptionalBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep,
 // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
(* <fileName>:<description> *)
if (<referencedElementsDeep:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator="
or ">) then
begin
 <alt>
end;
<referencedElementsDeep:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset;<\n>}>
>>

rewriteClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,     // elements in immediately block; no nested blocks
 description) ::=
<<
(* <fileName>:<description> *)
while (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">) do
begin
 <alt>
end;
<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset();<\n>}>
>>

rewritePositiveClosureBlock(
 alt,rewriteBlockLevel,
 referencedElementsDeep, // all nested refs
 referencedElements,
     // elements in immediately block; no nested blocks
 description) ::=
<<
if (not (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">)) then
 raise ERewriteEarlyExitException.Create('');

while (<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).HasNext}; separator=" or
">) do
begin
 <alt>
end;
<referencedElements:{el | (Locals['Stream_<el>'] as IRewriteRuleElementStream).Reset();<\n>}>
```

```
>>

rewriteAlt(a) ::= <<
(* <a.description> *)
<if(a.pred)>
if (<a.pred>) then
begin
 <a.alt>
end<\n>
<else>
begin
 <a.alt>
end;<\n>
<endif>
>>

/** For empty rewrites: "r : ... -> ;" */
rewriteEmptyAlt() ::= "Root[0] = null;"

rewriteTree(root,children,description,enclosingTreeLevel,treeLevel) ::= <<
(* <fileName>:<description> *)
begin
 Root[<treeLevel>] := Adaptor.GetNilNode as I<ASTLabelType>;
 <root:rewriteElement()>
 <children:rewriteElement()>
 Adaptor.AddChild(Root[<enclosingTreeLevel>],
 Root[<treeLevel>]);
end;<\n>
>>

rewriteElementList(elements) ::= "<elements:rewriteElement()>"

rewriteElement(e) ::= <<
<@pregen()>
<e.el>
>>

/** Gen ID or ID[args] */
rewriteTokenRef(token,elementIndex,hetero,args) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <createRewriteNodeFromElement(...)>);<\n>
>>

/** Gen $label ... where defined via label=ID */
rewriteTokenLabelRef(label,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as
IRewriteRuleElementStream).NextNode());<\n>
>>
```

```
/** Gen $label ... where defined via label+=ID */
rewriteTokenListLabelRef(label,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode);<\n>
>>

/** Gen ^($label ...) */
rewriteTokenLabelRefRoot(label,elementIndex) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as
IRewriteRuleElementStream).NextNode(), Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen ^($label ...) where label+=...
 */
rewriteTokenListLabelRefRoot ::= rewriteTokenLabelRefRoot

/** Gen ^(ID ...) or ^(ID[args] ...) */
rewriteTokenRefRoot(token,elementIndex,hetero,args) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<createRewriteNodeFromElement(...)>, Root[<treeLevel>]) as
I<ASTLabelType>;<\n>
>>

rewriteImaginaryTokenRef(args,token,hetero,elementIndex) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <createImaginaryNode(tokenType=token, ...)>);<\n>
>>

rewriteImaginaryTokenRefRoot(args,token,hetero,elementIndex) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<createImaginaryNode(tokenType=token, ...)>, Root[<treeLevel>]) as
I<ASTLabelType>;<\n>
>>

/** plain -> {foo} action */
rewriteAction(action) ::= <<
Root[0] = <action>;<\n>
>>

/** What is the name of the previous value of this rule's root tree?  This
*  let's us refer to $rule to mean previous value.  I am reusing the
*  variable 'tree' sitting in retval struct to hold the value of Root[0] right
*  before I set it during rewrites.  The assign
 will be to retval.Tree.
*/
prevRuleRootRef() ::= "RetVal"

rewriteRuleRef(rule) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<rule>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

rewriteRuleRefRoot(rule) ::= <<
```

```
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<rule>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

rewriteNodeAction(action) ::= <<
Adaptor.AddChild(Root[<treeLevel>], <action>);<\n>
>>

rewriteNodeActionRoot(action) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot(<action>, Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel=rule */
rewriteRuleLabelRef(label) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

/** Gen $ruleLabel ... where defined via ruleLabel+=rule */
rewriteRuleListLabelRef(label) ::= <<
Adaptor.AddChild(Root[<treeLevel>], (Locals['Stream_<label>'] as IRewriteRuleElementStream).NextTree());<\n>
>>

/**
 Gen ^($ruleLabel ...) where ruleLabel=rule */
rewriteRuleLabelRefRoot(label) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

/** Gen ^($ruleLabel ...) where ruleLabel+=rule */
rewriteRuleListLabelRefRoot(label) ::= <<
Root[<treeLevel>] := Adaptor.BecomeRoot((Locals['Stream_<label>'] as IRewriteRuleElementStream).NextNode,
Root[<treeLevel>]) as I<ASTLabelType>;<\n>
>>

createImaginaryNode(tokenType,hetero,args) ::= <<
<if(hetero)>
<! new MethodNode(IDLabel, args) !>
T<hetero>.Create(<tokenType><if(args)>, <args; separator=", "><endif>)
<else>
Adaptor.CreateNode(<tokenType>, <args; separator=", "><if(!args)>'<tokenType>'<endif>) as I<ASTLabelType>
<endif>
>>

createRewriteNodeFromElement(token,hetero,args) ::= <<
<if(hetero)>
T<hetero>.Create((Locals['Stream_<token>'] as IRewriteRuleElementStream).NextToken<if(args)>, <args;
separator=", "><endif>)
```

```
<else>
<if(args)>
 <! must create new node from old !>
Adaptor.Create(<token>, <args; separator=", ">)
<else>
(Locals['Stream_<token>'] as IRewriteRuleElementStream).NextNode
<endif>
<endif>
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Delphi/AST.stg
No license file was found, but licenses were detected in source scan.

```
group CSharp3 implements ANTLRCore;
```

```
csharpVisibilityMap ::= [
 "private":"private",
 "protected":"protected",
 "public":"public",
 "fragment":"private",
 default:"private"
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile( LEXER,PARSER,TREE_PARSER,
 actionScope, actions,
   docComment, recognizer,
   name, tokens, tokenNames, rules, cyclicDFAs,
   bitsets, buildTemplate, buildAST, rewriteMode, profile,
   backtracking, synpreds, memoize, numRules,
   fileName, ANTLRVersion, generatedTimestamp, trace,
   scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>

// The variable 'variable' is assigned but its value is never used.
#pragma warning disable 219
// Unreachable code detected.
#pragma warning disable 162

<actions.(actionScope).header>

<@imports>
using System.Collections.Generic;
using Antlr.Runtime;
<if(TREE_PARSER)>
using Antlr.Runtime.Tree;
using RewriteRuleITokenStream = Antlr.Runtime.Tree.RewriteRuleTokenStream;
<endif>
using Stack = System.Collections.Generic.Stack\<object>;
using List = System.Collections.IList;
using ArrayList = System.Collections.Generic.List\<object>;
<if(backtracking)>
using Map = System.Collections.IDictionary;
using HashMap = System.Collections.Generic.Dictionary\<object,
 object>;
<endif>
<@end>

<if(actions.(actionScope).namespace)>
```

```
namespace <actions.(actionScope).namespace>
{
<endif>

<docComment>
<recognizer>
<if(actions.(actionScope).namespace)>

} // namespace <actions.(actionScope).namespace>

<endif>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="IToken",
     filterMode, superClass="Lexer") ::= <<
public partial class <grammar.recognizerName> : <@superClassName><superClass><@end>
{
 <tokens:{public const int <it.name>=<it.type>;}; separator="\n">
 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <actions.lexer.members>

   // delegates
   <grammar.delegates:
        {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
   // delegators
   <grammar.delegators:
        {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
   <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

 public <grammar.recognizerName>() {}<! needed by subclasses
 !>
 public <grammar.recognizerName>( ICharStream input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}> )
  : this( input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}> )
 {
 }
 public <grammar.recognizerName>( ICharStream input, RecognizerSharedState state<grammar.delegators:{g|,
<g.recognizerName> <g:delegateName()>}> )
  : base( input, state )
 {
<if(memoize)>
<if(grammar.grammarIsRoot)>
  state.ruleMemo = new System.Collections.Generic.Dictionary\<int, int>[<numRules>+1];<\n><! index from 1..n !>
<endif>
<endif>
  <grammar.directDelegates:
   {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
```

```
  <grammar.delegators:
   {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
 }
 public override string GrammarFileName { get { return "<fileName>"; } }

<if(filterMode)>
 <filteringNextToken()>
<endif>
 <rules;
 separator="\n\n">

 <synpreds:{p | <lexerSynpred(p)>}>

 #region DFA
 <cyclicDFAs:{dfa | DFA<dfa.decisionNumber> dfa<dfa.decisionNumber>;}; separator="\n">

 protected override void InitDFAs()
 {
  base.InitDFAs();
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>( this<if(dfa.specialStateSTs)>,
new SpecialStateTransitionHandler( specialStateTransition<dfa.decisionNumber> )<endif> );}; separator="\n">
 }

 <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>
 #endregion

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public override IToken NextToken()
{
 for ( ; ;)
 {
  if ( input.LA(1)==CharStreamConstants.EOF )
  {
   return
 TokenConstants.EOF_TOKEN;
  }
  state.token = null;
  state.channel = TokenConstants.DEFAULT_CHANNEL;
  state.tokenStartCharIndex = input.Index;
  state.tokenStartCharPositionInLine = input.CharPositionInLine;
  state.tokenStartLine = input.Line;
  state.text = null;
  try
  {
   int m = input.Mark();
   state.backtracking=1;<! means we won't throw slow exception !>
   state.failed=false;
   mTokens();
   state.backtracking=0;
   <! mTokens backtracks with synpred at backtracking==2
      and we set the synpredgate to allow actions at level 1. !>
   if ( state.failed )
   {
    input.Rewind(m);
    input.Consume(); <! advance one char and try again !>
   }
   else
   {
    Emit();
    return state.token;
   }
  }
  catch ( RecognitionException re )
  {
   // shouldn't happen in backtracking mode, but...
   ReportError(re);
   Recover(re);
  }
 }
}

public override void Memoize( IIntStream input, int ruleIndex, int ruleStartIndex )
{
 if ( state.backtracking > 1 )
  base.Memoize( input,
 ruleIndex, ruleStartIndex );
}

public override bool AlreadyParsedRule(IIntStream input, int ruleIndex)
{
 if ( state.backtracking > 1 )
  return base.AlreadyParsedRule(input, ruleIndex);

 return false;
}
>>
```

```
actionGate() ::= "state.backtracking == 0"

filteringActionGate() ::= "state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
            ASTLabelType="object", labelType, members, rewriteElementType,
            filterMode) ::= <<
public partial class <grammar.recognizerName> : <@superClassName><superClass><@end>
{
<if(grammar.grammarIsRoot)>
 public static readonly string[] tokenNames = new string[] {
  "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
 };<\n>
<endif>
 <tokens:{public const int <it.name>=<it.type>;}; separator="\n">

 // delegates
 <grammar.delegates:
   {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
 //
 delegators
 <grammar.delegators:
   {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
 <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

 <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
 <@members>
<! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
public <grammar.recognizerName>( <inputStreamType> input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}> )
 : this( input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}> )
{
}
public <grammar.recognizerName>( <inputStreamType> input, RecognizerSharedState
state<grammar.delegators:{g|, <g.recognizerName> <g:delegateName()>}> )
 : base( input, state )
{
 <parserCtorBody()>
 <grammar.directDelegates:
  {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|, <p:delegateName()>}>,
this);}; separator="\n">
 <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
 separator="\n">
 <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
}
```

```
 <@end>

 public override string[] GetTokenNames() { return
<grammar.composite.rootGrammar.recognizerName>.tokenNames; }
 public override string GrammarFileName { get { return "<fileName>"; } }

 <members>

 #region Rules
 <rules; separator="\n\n">
 #endregion

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
 // Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
 public <returnType()> <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
<!throws RecognitionException !>\{ <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>}; separator=", ">); \}}; separator="\n">

 #region Synpreds
 <synpreds:{p | <synpred(p)>}>
 #endregion

 #region DFA
 <cyclicDFAs:{dfa | DFA<dfa.decisionNumber>
 dfa<dfa.decisionNumber>;}; separator="\n">

 protected override void InitDFAs()
 {
  base.InitDFAs();
  <cyclicDFAs:{dfa | dfa<dfa.decisionNumber> = new DFA<dfa.decisionNumber>( this<if(dfa.specialStateSTs)>,
new SpecialStateTransitionHandler( specialStateTransition<dfa.decisionNumber> )<endif> );}; separator="\n">
 }

 <cyclicDFAs:cyclicDFA()><! dump tables for all DFA !>
 #endregion

 #region Follow Sets
 public static class Follow
 {
  <bitsets:bitset(name={_<it.name>_in_<it.inName><it.tokenIndex>},
       words64=it.bits)>
 }
 #endregion
}
>>
```

```
parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new System.Collections.Generic.Dictionary\<int,
int>[<length(grammar.allImportedRules)>+1];<\n><! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
      ASTLabelType="object", superClass="Parser", labelType="IToken",
       members={<actions.parser.members>}) ::= <<
<genericParser(inputStreamType="ITokenStream", rewriteElementType="IToken", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
         numRules, bitsets, labelType={<ASTLabelType>}, ASTLabelType="object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
 members={<actions.treeparser.members>},
         filterMode) ::= <<
<genericParser(inputStreamType="ITreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates
 cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public <!final !>void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)<! throws
RecognitionException!>
{
 <ruleLabelDefs()>
<if(trace)>
 traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
 try
 {
  <block>
 }
 finally
 {
  traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
 }
<else>
 <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public <!final !>bool <name>()
{
 state.backtracking++;
 <@start()>
 int start = input.Mark();
 try
 {
  <name>_fragment(); // can never throw exception
 }
 catch ( RecognitionException re )
 {
  System.Console.Error.WriteLine("impossible: "+re);
 }
 bool success = !state.failed;
 input.Rewind(start);
 <@stop()>
 state.backtracking--;
 state.failed=false;
 return success;
}<\n>
>>

lexerSynpred(name) ::=
 <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( state.backtracking>0 && AlreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>; }
<endif>
>>

/** How to test for failure and return from rule */
```

```
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
<csharpVisibilityMap.(ruleDescriptor.modifier)> <returnType()>
 <ruleName>( <ruleDescriptor.parameterScope:parameterScope(scope=it)> )<! throws RecognitionException!>
{
 <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
 <ruleScopeSetUp()>
 <ruleDeclarations()>
 <ruleLabelDefs()>
 <ruleDescriptor.actions.init>
 <@preamble()>
 try
 {
  <ruleMemoization(name=ruleName)>
  <block>
  <ruleCleanUp()>
  <(ruleDescriptor.actions.after):execAction()>
 }
<if(exceptions)>
 <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
 <actions.(actionScope).rulecatch>
<else>
 catch ( RecognitionException re )
 {
  ReportError(re);
  Recover(input,re);
 <@setErrorReturnValue()>
 }<\n>
<endif>
<endif>
<endif>
 finally
 {
  <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
  <memoize()>
  <ruleScopeCleanUp()>
  <finally>
 }
 <@postamble()>
 return <ruleReturnValue()>;
}
// $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch ( <e.decl> )
{
 <e.action>
}
>>

ruleDeclarations()
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = new <returnType()>();
retval.start = input.LT(1);<\n>
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.Index;
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>_scope.PushScope(this);}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_scope.PushScope(this);}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_scope.PopScope(this);}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_scope.PopScope(this);}; separator="\n">
>>

ruleLabelDefs() ::= <<
```

```
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
  :{<labelType> <it.label.text>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{List list_<it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:ruleLabelDef(label=it); separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
  :{<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
  :{List list_<it.label.text>=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if
```

```
 ( state.backtracking>0 ) { Memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
<csharpVisibilityMap.(ruleDescriptor.modifier)> void
m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
{
 <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
 <ruleScopeSetUp()>
 <ruleDeclarations()>
 try
 {
<if(nakedBlock)>
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block><\n>
<else>
  int _type = <ruleName>;
  int _channel = DEFAULT_TOKEN_CHANNEL;
  <ruleMemoization(name=ruleName)>
  <lexerRuleLabelDefs()>
  <ruleDescriptor.actions.init>
  <block>
  <ruleCleanUp()>
  state.type = _type;
  state.channel = _channel;
  <(ruleDescriptor.actions.after):execAction()>
<endif>
 }
 finally
 {
  <if(trace)>traceOut("<ruleName>",
 <ruleDescriptor.index>);<endif>
  <ruleScopeCleanUp()>
  <memoize()>
 }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
```

```
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
public override void mTokens()<! throws RecognitionException!>
{
 <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch ( alt<decisionNumber> )
{
<alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
switch
 ( alt<decisionNumber> )
{
<alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
```

```
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int cnt<decisionNumber>=0;
<decls>
<@preloop()>
for ( ; ; )
{
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch ( alt<decisionNumber> )
 {
 <alts:altSwitchCase()>
 default:
  if ( cnt<decisionNumber> >= 1 )
   goto loop<decisionNumber>;

  <ruleBacktrackFailure()>
  EarlyExitException
 eee<decisionNumber> = new EarlyExitException( <decisionNumber>, input );
  <@earlyExitException()>
  throw eee<decisionNumber>;
 }
 cnt<decisionNumber>++;
}
loop<decisionNumber>:
 ;

<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
```

```
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
for ( ; ; )
{
 int alt<decisionNumber>=<maxAlt>;
 <@predecision()>
 <decision>
 <@postdecision()>
 switch ( alt<decisionNumber> )
 {
 <alts:altSwitchCase()>
 default:
  goto loop<decisionNumber>;
 }
}

loop<decisionNumber>:
 ;

<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/**
 A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i>:
 <@prealt()>
 <it>
 break;<\n>
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
```

```
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

// E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label>=(<labelType>)<endif>Match(input,<token>,Follow._<token>_in_<ruleName><elementIndex>);
<checkRuleBacktrackFailure()>
>>

/**
 ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList();
list_<label>.Add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
Match(<char>); <checkRuleBacktrackFailure()>
>>
```

```
/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
MatchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<endif>
if ( <s> )
{
 input.Consume();
 <postmatchCode>
<if(!LEXER)>
 state.errorRecovery=false;
<endif>
 <if(backtracking)>state.failed=false;<endif>
}
else
{
 <ruleBacktrackFailure()>
 MismatchedSetException mse =
 new MismatchedSetException(null,input);
 <@mismatchedSetException()>
<if(LEXER)>
 Recover(mse);
 throw mse;
<else>
 throw mse;
 <! use following code to make it recover inline; remove throw mse;
 recoverFromMismatchedSet(input,mse,Follow._set_in_<ruleName><elementIndex>);
 !>
<endif>
}<\n>
>>

matchSetUnchecked(s,label,elementIndex,postmatchCode="") ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
```

```
<else>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<endif>
input.Consume();
<postmatchCode>
<if(!LEXER)>
 state.errorRecovery=false;
<endif>
 <if(backtracking)>state.failed=false;<endif>
>>

matchRuleBlockSet ::= matchSet

matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
int <label>Start = GetCharIndex();
Match(<string>); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL,
 <label>Start, GetCharIndex()-1);
<else>
Match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
MatchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
```

```
<endif>
MatchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
ruleRef(rule,label,elementIndex,args,scope) ::= <<
PushFollow(Follow._<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args;
 separator=", ">);<\n>
state._fsp--;
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = GetCharIndex();
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = new CommonToken(input, TokenConstants.INVALID_TOKEN_TYPE,
TokenConstants.DEFAULT_CHANNEL, <label>Start<elementIndex>, GetCharIndex()-1);
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>
```

```
/**
 i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = GetCharIndex();
Match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, TokenConstants.DEFAULT_CHANNEL,
<label>Start<elementIndex>, GetCharIndex()-1);
<else>
Match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==TokenConstants.DOWN )
{
 Match(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
 <children:element()>
 Match(input, TokenConstants.UP, null); <checkRuleBacktrackFailure()>
}
<else>
Match(input, TokenConstants.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
Match(input,
 TokenConstants.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) )
{
 <ruleBacktrackFailure()>
 throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
```

```
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else
{
<if(eotPredictsAlt)>
 alt<decisionNumber>=<eotPredictsAlt>;
<else>
 <ruleBacktrackFailure()>
 NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>, <stateNumber>,
input);<\n>
 <@noViableAltException()>
 throw nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.  It delays
 error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else
{
 alt<decisionNumber>=<eotPredictsAlt>;
```

```
}<\n>
<endif>
<endif>
>>

/**
 An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>)
{
 <targetState>
}
>>

// F i x e d  D F A  (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
 alt<decisionNumber>=<eotPredictsAlt>;
 break;<\n>
<else>
 {
  <ruleBacktrackFailure()>
  NoViableAltException nvae = new NoViableAltException("<description>", <decisionNumber>,
 <stateNumber>, input);<\n>
  <@noViableAltException()>
  throw nvae;
 }<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
```

```
<edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) )
{
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
 alt<decisionNumber>=<eotPredictsAlt>;
 break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n">
 {
 <targetState>
 }
 break;
>>

// C y c l i c  D F A

/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.Predict(input);
>>

/* Dump DFA tables as run-length-encoded Strings
 of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
class DFA<dfa.decisionNumber> : DFA
{

 const string DFA<dfa.decisionNumber>_eotS =
  "<dfa.javaCompressedEOT; wrap="\"+\n\t\t\"">";
 const string DFA<dfa.decisionNumber>_eofS =
  "<dfa.javaCompressedEOF; wrap="\"+\n\t\t\"">";
 const string DFA<dfa.decisionNumber>_minS =
  "<dfa.javaCompressedMin; wrap="\"+\n\t\t\"">";
 const string DFA<dfa.decisionNumber>_maxS =
  "<dfa.javaCompressedMax; wrap="\"+\n\t\t\"">";
 const string DFA<dfa.decisionNumber>_acceptS =
  "<dfa.javaCompressedAccept; wrap="\"+\n\t\t\"">";
 const string DFA<dfa.decisionNumber>_specialS =
  "<dfa.javaCompressedSpecial; wrap="\"+\n\t\t\"">}>";
 static
 readonly string[] DFA<dfa.decisionNumber>_transitionS =
  {
   <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
  };

 static readonly short[] DFA<dfa.decisionNumber>_eot =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eotS);
 static readonly short[] DFA<dfa.decisionNumber>_eof =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_eofS);
 static readonly char[] DFA<dfa.decisionNumber>_min =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
 static readonly char[] DFA<dfa.decisionNumber>_max =
DFA.UnpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
 static readonly short[] DFA<dfa.decisionNumber>_accept =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
 static readonly short[] DFA<dfa.decisionNumber>_special =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_specialS);
 static readonly short[][] DFA<dfa.decisionNumber>_transition;

 static DFA<dfa.decisionNumber>()
 {
  int numStates = DFA<dfa.decisionNumber>_transitionS.Length;
  DFA<dfa.decisionNumber>_transition
 = new short[numStates][];
  for ( int i=0; i \< numStates; i++ )
  {
   DFA<dfa.decisionNumber>_transition[i] =
DFA.UnpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
  }
 }

 public DFA<dfa.decisionNumber>( BaseRecognizer recognizer<if(dfa.specialStateSTs)>,
SpecialStateTransitionHandler specialStateTransition<endif> )
<if(dfa.specialStateSTs)>
  : base( specialStateTransition )
<endif>
```

```
 {
  this.recognizer = recognizer;
  this.decisionNumber = <dfa.decisionNumber>;
  this.eot = DFA<dfa.decisionNumber>_eot;
  this.eof = DFA<dfa.decisionNumber>_eof;
  this.min = DFA<dfa.decisionNumber>_min;
  this.max = DFA<dfa.decisionNumber>_max;
  this.accept = DFA<dfa.decisionNumber>_accept;
  this.special = DFA<dfa.decisionNumber>_special;
  this.transition = DFA<dfa.decisionNumber>_transition;
 }
 public override string GetDescription()
 {
  return "<dfa.description>";
 }
 <@errorMethod()>
}<\n>
<if(dfa.specialStateSTs)>
int specialStateTransition<dfa.decisionNumber>(
 DFA dfa, int s, IIntStream _input )<! throws NoViableAltException!>
{
 <if(LEXER)>
 IIntStream input = _input;
 <endif>
 <if(PARSER)>
 ITokenStream input = (ITokenStream)_input;
 <endif>
 <if(TREE_PARSER)>
 ITreeNodeStream input = (ITreeNodeStream)_input;
 <endif>
 int _s = s;
 switch ( s )
 {
 <dfa.specialStateSTs:{state |
 case <i0>:<! compressed special state numbers 0..n-1 !>
  <state>}; separator="\n">
 }
<if(backtracking)>
 if (state.backtracking>0) {state.failed=true; return -1;}<\n>
<endif>
 NoViableAltException nvae = new NoViableAltException(dfa.GetDescription(), <dfa.decisionNumber>, _s, input);
 dfa.Error(nvae);
 throw nvae;
}<\n>
<endif>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
```

```
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(1);<\n>
<if(semPredState)><! get next lookahead symbol to test edges, then
 rewind !>
int index<decisionNumber>_<stateNumber> = input.Index;
input.Rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)><! return input cursor to state before we rewound !>
input.Seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"

orPredicates(operands) ::= "(<first(operands)><rest(operands):{o | ||<o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description)
 ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"
```

```
/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber>\>=<lower> && LA<decisionNumber>_<stateNumber>\<=<upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>)\>=<lower>
&& input.LA(<k>)\<=<upper>)"

setTest(ranges) ::= "<ranges; separator=\"||\">"

// A T T R I B U T E S

attributeScope(scope) ::= <<
<if(scope.attributes)>
protected class <scope.name>_scope
{
<if(scope.actions.scopeinit)>
 public <scope.name>_scope()
 {
  <scope.actions.scopeinit>
 }<\n>
<endif>
<if(scope.actions.scopeafter)>
 void CleanUpScope()
 {
  <scope.actions.scopeafter>
 }<\n>
<endif>
 public
 static void PushScope( <scope.grammar.recognizerName> grammar )
 {
  grammar.<scope.name>_stack.Push( new <scope.name>_scope() );
 }
 public static void PopScope( <scope.grammar.recognizerName> grammar )
 {
  grammar.<scope.name>_stack.Pop()<if(scope.actions.scopeafter)>.CleanUpScope()<endif>;
 }

 <scope.attributes:{public <it.decl>;}; separator="\n">
}
protected Stack\<<scope.name>_scope\> <scope.name>_stack = new Stack\<<scope.name>_scope\>();<\n>
<endif>
>>
```

```
globalAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

ruleAttributeScope(scope) ::= <<
<attributeScope(...)>
>>

returnStructName() ::= "<it.name>_return"

returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the C# type associated with a single or
 multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>.<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/** Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
default(<typeName>)
>>
```

```
/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public
 <!static !>class <ruleDescriptor:returnStructName()> :
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope
{
 <scope.attributes:{public <it.decl>;}; separator="\n">
 <@ruleReturnMembers()>
}
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr) ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
<else>
((<scope>_scope)<scope>_stack.Peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name> =<expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
```

```
 =<expr>;
<else>
((<scope>_scope)<scope>_stack.Peek()).<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"

/** reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?<scope>.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>;
<else>
<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID
 {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"


// not sure the next are the right approach
```

```
tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.Text:null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.Type:0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.Line:0)"
tokenLabelPropertyRef_pos(scope,attr) ::= "(<scope>!=null?<scope>.CharPositionInLine:0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.Channel:0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.TokenIndex:0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?Integer.valueOf(<scope>.Text):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.start):null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.stop):null)"
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>!=null?((<ASTLabelType>)<scope>.tree):null)"
ruleLabelPropertyRef_text(scope,attr)
 ::= <<
<if(TREE_PARSER)>
(<scope>!=null?(input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(<scope>.start),
 input.TreeAdaptor.GetTokenStopIndex(<scope>.start))):null)
<else>
(<scope>!=null?input.ToString(<scope>.start,<scope>.stop):null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr) ::= "(<scope>!=null?<scope>.st:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(<scope>!=null?<scope>.Type:0)"

lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(<scope>!=null?<scope>.Line:0)"

lexerRuleLabelPropertyRef_pos(scope,attr) ::=
   "(<scope>!=null?<scope>.CharPositionInLine:-1)"

lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(<scope>!=null?<scope>.Channel:0)"

lexerRuleLabelPropertyRef_index(scope,attr) ::=
   "(<scope>!=null?<scope>.TokenIndex:0)"

lexerRuleLabelPropertyRef_text(scope,attr)
 ::=
   "(<scope>!=null?<scope>.Text:null)"

lexerRuleLabelPropertyRef_int(scope,attr) ::=
   "(<scope>!=null?Integer.valueOf(<scope>.Text):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.start)"
rulePropertyRef_stop(scope,attr) ::= "((<labelType>)retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.TokenStream.ToString(
 input.TreeAdaptor.GetTokenStartIndex(retval.start),
 input.TreeAdaptor.GetTokenStopIndex(retval.start))
<else>
input.ToString(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"

lexerRulePropertyRef_text(scope,attr) ::= "Text"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::=
 "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(GetCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr) ::= "int.Parse(<scope>.Text)"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> )
{
 <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"
```

```
// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static readonly BitSet <name> = new BitSet(new ulong[]{<words64:{<it>UL};separator=",">});<\n>
>>

codeFileExtension()
 ::= ".cs"

true() ::= "true"
false() ::= "false"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/CSharp3/CSharp3.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]
Copyright (c) 2005-2006 Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
group Java implements ANTLRCore;

javaTypeInitMap ::= [
 "int":"0",
 "long":"0",
 "float":"0.0f",
 "double":"0.0",
 "boolean":"false",
 "byte":"0",
 "short":"0",
 "char":"0",
 default:"null" // anything other than an atomic type
]

/** The overall file structure of a recognizer; stores methods for rules
*  and cyclic DFAs plus support code.
*/
outputFile(LEXER,PARSER,TREE_PARSER, actionScope, actions,
           docComment, recognizer,
           name, tokens, tokenNames, rules,
 cyclicDFAs,
    bitsets, buildTemplate, buildAST, rewriteMode, profile,
    backtracking, synpreds, memoize, numRules,
    fileName, ANTLRVersion, generatedTimestamp, trace,
    scopes, superClass, literals) ::=
<<
// $ANTLR <ANTLRVersion> <fileName> <generatedTimestamp>
<actions.(actionScope).header>

<@imports>
import org.antlr.runtime.*;
<if(TREE_PARSER)>
import org.antlr.runtime.tree.*;
<endif>
import java.util.Stack;
import java.util.List;
import java.util.ArrayList;
<if(backtracking)>
import java.util.Map;
import java.util.HashMap;
<endif>
<@end>

<docComment>
<recognizer>
>>

lexer(grammar, name, tokens, scopes, rules, numRules, labelType="Token",
     filterMode, superClass="Lexer") ::= <<
public class <grammar.recognizerName> extends <@superClassName><superClass><@end> {
   <tokens:{public static final int <it.name>=<it.type>;}; separator="\n">
```

```
  <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
  <actions.lexer.members>

  // delegates
   <grammar.delegates:
      {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  // delegators
  <grammar.delegators:
      {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

  public <grammar.recognizerName>() {;} <! needed by subclasses !>
  public <grammar.recognizerName>(CharStream input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>) {
     this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>);
  }
  public <grammar.recognizerName>(CharStream input, RecognizerSharedState state<grammar.delegators:{g|,
<g.recognizerName> <g:delegateName()>}>) {
     super(input,state);
<if(memoize)>
<if(grammar.grammarIsRoot)>
     state.ruleMemo = new HashMap[<numRules>+1];<\n> <! index from 1..n !>
<endif>
<endif>
     <grammar.directDelegates:
      {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
 <p:delegateName()>}>, this);}; separator="\n">
     <grammar.delegators:
      {g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
     <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
  }
  public String getGrammarFileName() { return "<fileName>"; }

<if(filterMode)>
  <filteringNextToken()>
<endif>
  <rules; separator="\n\n">

  <synpreds:{p | <lexerSynpred(p)>}>

  <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
  <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

}
>>

/** A override of Lexer.nextToken() that backtracks over mTokens() looking
```

```
*  for matches.  No error can be generated upon error; just rewind, consume
*  a token and then try again.  backtracking needs to be set as well.
*  Make rule memoization happen only at levels above 1 as we start mTokens
*  at backtracking==1.
*/
filteringNextToken() ::= <<
public Token
 nextToken() {
   while (true) {
       if ( input.LA(1)==CharStream.EOF ) {
          return Token.EOF_TOKEN;
       }
       state.token = null;
 state.channel = Token.DEFAULT_CHANNEL;
       state.tokenStartCharIndex = input.index();
       state.tokenStartCharPositionInLine = input.getCharPositionInLine();
       state.tokenStartLine = input.getLine();
 state.text = null;
       try {
          int m = input.mark();
          state.backtracking=1; <! means we won't throw slow exception !>
          state.failed=false;
          mTokens();
          state.backtracking=0;
          <! mTokens backtracks with synpred at backtracking==2
            and we set the synpredgate to allow actions at level 1. !>
          if ( state.failed ) {
             input.rewind(m);
             input.consume(); <! advance one char and try again !>
          }
          else {
             emit();
             return state.token;
          }
       }
        catch (RecognitionException re) {
          // shouldn't happen in backtracking mode, but...
          reportError(re);
          recover(re);
       }
   }
}

public void memoize(IntStream input,
  int ruleIndex,
  int ruleStartIndex)
{
if ( state.backtracking>1 ) super.memoize(input, ruleIndex, ruleStartIndex);
```

```
}

public boolean alreadyParsedRule(IntStream input, int ruleIndex) {
if ( state.backtracking>1 ) return super.alreadyParsedRule(input, ruleIndex);
return false;
}
>>

actionGate() ::= "state.backtracking==0"

filteringActionGate() ::= "state.backtracking==1"

/** How to generate a parser */
genericParser(grammar, name, scopes, tokens, tokenNames, rules, numRules,
            bitsets, inputStreamType, superClass,
            ASTLabelType="Object", labelType, members, rewriteElementType,
            filterMode) ::= <<
public class <grammar.recognizerName> extends <@superClassName><superClass><@end> {
<if(grammar.grammarIsRoot)>
  public static final String[]
 tokenNames = new String[] {
     "\<invalid>", "\<EOR>", "\<DOWN>", "\<UP>", <tokenNames; separator=", ">
  };<\n>
<endif>
  <tokens:{public static final int <it.name>=<it.type>;}; separator="\n">

  // delegates
  <grammar.delegates:
      {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  // delegators
  <grammar.delegators:
      {g|public <g.recognizerName> <g:delegateName()>;}; separator="\n">
  <last(grammar.delegators):{g|public <g.recognizerName> gParent;}>

  <scopes:{<if(it.isDynamicGlobalScope)><globalAttributeScope(scope=it)><endif>}>
  <@members>
  <! WARNING. bug in ST: this is cut-n-paste into Dbg.stg !>
  public <grammar.recognizerName>(<inputStreamType> input<grammar.delegators:{g|, <g.recognizerName>
<g:delegateName()>}>) {
      this(input, new RecognizerSharedState()<grammar.delegators:{g|, <g:delegateName()>}>);
  }
  public <grammar.recognizerName>(<inputStreamType> input, RecognizerSharedState
state<grammar.delegators:{g|,
 <g.recognizerName> <g:delegateName()>}>) {
      super(input, state);
      <parserCtorBody()>
      <grammar.directDelegates:
       {g|<g:delegateName()> = new <g.recognizerName>(input, state<trunc(g.delegators):{p|,
<p:delegateName()>}>, this);}; separator="\n">
      <grammar.indirectDelegates:{g | <g:delegateName()> = <g.delegator:delegateName()>.<g:delegateName()>;};
separator="\n">
      <last(grammar.delegators):{g|gParent = <g:delegateName()>;}>
   }
   <@end>

   public String[] getTokenNames() { return <grammar.composite.rootGrammar.recognizerName>.tokenNames; }
   public String getGrammarFileName() { return "<fileName>"; }

   <members>

   <rules; separator="\n\n">

<! generate rule/method definitions for imported rules so they
  appear to be defined in this recognizer. !>
   // Delegated rules
<grammar.delegatedRules:{ruleDescriptor|
   public <returnType()> <ruleDescriptor.name>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>)
 throws RecognitionException \{ <if(ruleDescriptor.hasReturnValue)>return
<endif><ruleDescriptor.grammar:delegateName()>.<ruleDescriptor.name>(<ruleDescriptor.parameterScope.attribut
es:{a|<a.name>}; separator=", ">); \}}; separator="\n">

   <synpreds:{p | <synpred(p)>}>

   <cyclicDFAs:{dfa | protected DFA<dfa.decisionNumber> dfa<dfa.decisionNumber> = new
DFA<dfa.decisionNumber>(this);}; separator="\n">
   <cyclicDFAs:cyclicDFA()> <! dump tables for all DFA !>

   <bitsets:bitset(name={FOLLOW_<it.name>_in_<it.inName><it.tokenIndex>},
                  words64=it.bits)>
}
>>

parserCtorBody() ::= <<
<if(memoize)>
<if(grammar.grammarIsRoot)>
this.state.ruleMemo = new HashMap[<length(grammar.allImportedRules)>+1];<\n> <! index from 1..n !>
<endif>
<endif>
<grammar.delegators:
{g|this.<g:delegateName()> = <g:delegateName()>;}; separator="\n">
>>

parser(grammar, name, scopes, tokens, tokenNames, rules, numRules, bitsets,
      ASTLabelType="Object", superClass="Parser",
 labelType="Token",
      members={<actions.parser.members>}) ::= <<
```

```
<genericParser(inputStreamType="TokenStream", rewriteElementType="Token", ...)>
>>

/** How to generate a tree parser; same as parser except the input
*  stream is a different type.
*/
treeParser(grammar, name, scopes, tokens, tokenNames, globalAction, rules,
           numRules, bitsets, labelType={<ASTLabelType>}, ASTLabelType="Object",
superClass={<if(filterMode)><if(buildAST)>TreeRewriter<else>TreeFilter<endif><else>TreeParser<endif>},
 members={<actions.treeparser.members>},
           filterMode) ::= <<
<genericParser(inputStreamType="TreeNodeStream", rewriteElementType="Node", ...)>
>>

/** A simpler version of a rule template that is specific to the imaginary
*  rules created for syntactic predicates.  As they never have return values
*  nor parameters etc..., just give simplest possible method.  Don't do
*  any of the normal memoization stuff in here either; it's a waste.
*  As predicates
 cannot be inlined into the invoking rule, they need to
*  be in a rule by themselves.
*/
synpredRule(ruleName, ruleDescriptor, block, description, nakedBlock) ::=
<<
// $ANTLR start <ruleName>
public final void <ruleName>_fragment(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) throws
RecognitionException {
   <ruleLabelDefs()>
<if(trace)>
   traceIn("<ruleName>_fragment", <ruleDescriptor.index>);
   try {
       <block>
   }
   finally {
       traceOut("<ruleName>_fragment", <ruleDescriptor.index>);
   }
<else>
   <block>
<endif>
}
// $ANTLR end <ruleName>
>>

synpred(name) ::= <<
public final boolean <name>() {
   state.backtracking++;
   <@start()>
   int start = input.mark();
```

```
    try {
        <name>_fragment(); // can never throw exception
    } catch (RecognitionException re) {
        System.err.println("impossible: "+re);
    }
    boolean success = !state.failed;
    input.rewind(start);
    <@stop()>
    state.backtracking--;
    state.failed=false;
     return success;
}<\n>
>>

lexerSynpred(name) ::= <<
<synpred(name)>
>>

ruleMemoization(name) ::= <<
<if(memoize)>
if ( state.backtracking>0 && alreadyParsedRule(input, <ruleDescriptor.index>) ) { return <ruleReturnValue()>; }
<endif>
>>

/** How to test for failure and return from rule */
checkRuleBacktrackFailure() ::= <<
<if(backtracking)>if (state.failed) return <ruleReturnValue()>;<endif>
>>

/** This rule has failed, exit indicating failure during backtrack */
ruleBacktrackFailure() ::= <<
<if(backtracking)>if (state.backtracking>0) {state.failed=true; return <ruleReturnValue()>;}<endif>
>>

/** How to generate code for a rule.  This includes any return type
*  data aggregates required for multiple return values.
*/
rule(ruleName,ruleDescriptor,block,emptyRule,description,exceptions,finally,memoize) ::= <<
<ruleAttributeScope(scope=ruleDescriptor.ruleScope)>
<returnScope(scope=ruleDescriptor.returnScope)>

// $ANTLR start "<ruleName>"
// <fileName>:<description>
public
 final <returnType()> <ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) throws
RecognitionException {
    <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeSetUp()>
```

```
   <ruleDeclarations()>
   <ruleLabelDefs()>
   <ruleDescriptor.actions.init>
   <@preamble()>
   try {
      <ruleMemoization(name=ruleName)>
      <block>
      <ruleCleanUp()>
      <(ruleDescriptor.actions.after):execAction()>
   }
<if(exceptions)>
   <exceptions:{e|<catch(decl=e.decl,action=e.action)><\n>}>
<else>
<if(!emptyRule)>
<if(actions.(actionScope).rulecatch)>
   <actions.(actionScope).rulecatch>
<else>
   catch (RecognitionException re) {
      reportError(re);
      recover(input,re);
 <@setErrorReturnValue()>
   }<\n>
<endif>
<endif>
<endif>
   finally {
      <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
      <memoize()>
      <ruleScopeCleanUp()>
      <finally>
   }
   <@postamble()>
   return <ruleReturnValue()>;
}
//
 $ANTLR end "<ruleName>"
>>

catch(decl,action) ::= <<
catch (<e.decl>) {
   <e.action>
}
>>

ruleDeclarations() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<returnType()> retval = new <returnType()>();
retval.start = input.LT(1);<\n>
```

```
<else>
<ruleDescriptor.returnScope.attributes:{ a |
<a.type> <a.name> = <if(a.initValue)><a.initValue><else><initValue(a.type)><endif>;
}>
<endif>
<if(memoize)>
int <ruleDescriptor.name>_StartIndex = input.index();
<endif>
>>

ruleScopeSetUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.push(new <it>_scope());}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.push(new <it.name>_scope());}; separator="\n">
>>

ruleScopeCleanUp() ::= <<
<ruleDescriptor.useScopes:{<it>_stack.pop();}; separator="\n">
<ruleDescriptor.ruleScope:{<it.name>_stack.pop();}; separator="\n">
>>

ruleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,ruleDescriptor.tokenListLabels,
 ruleDescriptor.wildcardTreeLabels,ruleDescriptor.wildcardTreeListLabels]
    :{<labelType> <it.label.text>=null;}; separator="\n"
>
<[ruleDescriptor.tokenListLabels,ruleDescriptor.ruleListLabels,ruleDescriptor.wildcardTreeListLabels]
   :{List list_<it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.ruleLabels:ruleLabelDef(label=it); separator="\n">
<ruleDescriptor.ruleListLabels:{ll|<ll:ruleLabelDef(label=it)> <ll.label.text> = null;}; separator="\n">
>>

lexerRuleLabelDefs() ::= <<
<[ruleDescriptor.tokenLabels,
 ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleLabels]
   :{<labelType> <it.label.text>=null;}; separator="\n"
>
<ruleDescriptor.charLabels:{int <it.label.text>;}; separator="\n">
<[ruleDescriptor.tokenListLabels,
 ruleDescriptor.ruleListLabels]
   :{List list_<it.label.text>=null;}; separator="\n"
>
>>

ruleReturnValue() ::= <<
<if(!ruleDescriptor.isSynPred)>
<if(ruleDescriptor.hasReturnValue)>
```

```
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnName>
<else>
retval
<endif>
<endif>
<endif>
>>

ruleCleanUp()
 ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<if(!TREE_PARSER)>
retval.stop = input.LT(-1);<\n>
<endif>
<endif>
>>

memoize() ::= <<
<if(memoize)>
<if(backtracking)>
if ( state.backtracking>0 ) { memoize(input, <ruleDescriptor.index>, <ruleDescriptor.name>_StartIndex); }
<endif>
<endif>
>>

/** How to generate a rule in the lexer; naked blocks are used for
*  fragment rules.
*/
lexerRule(ruleName,nakedBlock,ruleDescriptor,block,memoize) ::= <<
// $ANTLR start "<ruleName>"
public final void m<ruleName>(<ruleDescriptor.parameterScope:parameterScope(scope=it)>) throws
RecognitionException {
   <if(trace)>traceIn("<ruleName>", <ruleDescriptor.index>);<endif>
   <ruleScopeSetUp()>
   <ruleDeclarations()>
   try {
<if(nakedBlock)>
       <ruleMemoization(name=ruleName)>
       <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
       <block><\n>
<else>
       int _type = <ruleName>;
       int _channel = DEFAULT_TOKEN_CHANNEL;
       <ruleMemoization(name=ruleName)>
        <lexerRuleLabelDefs()>
       <ruleDescriptor.actions.init>
```

```
    <block>
    <ruleCleanUp()>
    state.type = _type;
    state.channel = _channel;
    <(ruleDescriptor.actions.after):execAction()>
<endif>
  }
  finally {
    <if(trace)>traceOut("<ruleName>", <ruleDescriptor.index>);<endif>
    <ruleScopeCleanUp()>
    <memoize()>
  }
}
// $ANTLR end "<ruleName>"
>>

/** How to generate code for the implicitly-defined lexer grammar rule
*  that chooses between lexer rules.
*/
tokensRule(ruleName,nakedBlock,args,block,ruleDescriptor) ::= <<
public void mTokens() throws RecognitionException {
  <block><\n>
}
>>

// S U B R U L E S

/** A (...) subrule with multiple alternatives */
block(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
<@predecision()>
<decision>
<@postdecision()>
<@prebranch()>
switch (alt<decisionNumber>)
 {
  <alts:altSwitchCase()>
}
<@postbranch()>
>>

/** A rule block with multiple alternatives */
ruleBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::= <<
// <fileName>:<description>
int alt<decisionNumber>=<maxAlt>;
<decls>
```

```
<@predecision()>
<decision>
<@postdecision()>
switch (alt<decisionNumber>) {
   <alts:altSwitchCase()>
}
>>

ruleBlockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A special case of a (...) subrule with a single alternative */
blockSingleAlt(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,description) ::= <<
// <fileName>:<description>
<decls>
<@prealt()>
<alts>
<@postalt()>
>>

/** A (..)+ block with 1 or more alternatives */
positiveClosureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,descriptio
n) ::= <<
// <fileName>:<description>
int
 cnt<decisionNumber>=0;
<decls>
<@preloop()>
loop<decisionNumber>:
do {
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default :
     if ( cnt<decisionNumber> >= 1 ) break loop<decisionNumber>;
     <ruleBacktrackFailure()>
           EarlyExitException eee =
               new EarlyExitException(<decisionNumber>, input);
           <@earlyExitException()>
           throw eee;
   }
   cnt<decisionNumber>++;
} while (true);
<@postloop()>
>>

positiveClosureBlockSingleAlt ::= positiveClosureBlock

/** A (..)* block with 1 or more alternatives */
closureBlock(alts,decls,decision,enclosingBlockLevel,blockLevel,decisionNumber,maxK,maxAlt,description) ::=
<<
// <fileName>:<description>
<decls>
<@preloop()>
loop<decisionNumber>:
do {
   int alt<decisionNumber>=<maxAlt>;
   <@predecision()>
   <decision>
   <@postdecision()>
   switch (alt<decisionNumber>) {
 <alts:altSwitchCase()>
 default
 :
     break loop<decisionNumber>;
   }
} while (true);
<@postloop()>
>>

closureBlockSingleAlt ::= closureBlock

/** Optional blocks (x)? are translated to (x|) by before code generation
*  so we can just use the normal block template
*/
optionalBlock ::= block

optionalBlockSingleAlt ::= block

/** A case in a switch that jumps to an alternative given the alternative
*  number.  A DFA predicts the alternative and then a simple switch
*  does the jump to the code that actually matches that alternative.
*/
altSwitchCase() ::= <<
case <i> :
   <@prealt()>
   <it>
   break;<\n>
```

```
>>

/** An alternative is just a list of elements; at outermost level */
alt(elements,altNum,description,autoAST,outerAlt,treeLevel,rew) ::= <<
// <fileName>:<description>
{
<@declarations()>
<elements:element()>
<rew>
<@cleanup()>
}
>>

/** What to emit when there is no rewrite.  For auto build
*  mode, does nothing.
*/
noRewrite(rewriteBlockLevel, treeLevel) ::= ""

//
 E L E M E N T S

/** Dump the elements one per line */
element() ::= <<
<@prematch()>
<it.el><\n>
>>

/** match a token optionally with a label in front */
tokenRef(token,label,elementIndex,hetero) ::= <<
<if(label)><label>=(<labelType>)<endif>match(input,<token>,FOLLOW_<token>_in_<ruleName><elementIndex
>); <checkRuleBacktrackFailure()>
>>

/** ids+=ID */
tokenRefAndListLabel(token,label,elementIndex,hetero) ::= <<
<tokenRef(...)>
<listLabel(elem=label,...)>
>>

listLabel(label,elem) ::= <<
if (list_<label>==null) list_<label>=new ArrayList();
list_<label>.add(<elem>);<\n>
>>

/** match a character */
charRef(char,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
```

```
<endif>
match(<char>); <checkRuleBacktrackFailure()>
>>

/** match a character range */
charRangeRef(a,b,label) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchRange(<a>,<b>); <checkRuleBacktrackFailure()>
>>

/** For now, sets are interval tests and must be tested inline */
matchSet(s,label,elementIndex,postmatchCode="")
 ::= <<
<if(label)>
<if(LEXER)>
<label>= input.LA(1);<\n>
<else>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
<endif>
if ( <s> ) {
   input.consume();
   <postmatchCode>
<if(!LEXER)>
   state.errorRecovery=false;
<endif>
   <if(backtracking)>state.failed=false;<endif>
}
else {
   <ruleBacktrackFailure()>
   MismatchedSetException mse = new MismatchedSetException(null,input);
   <@mismatchedSetException()>
<if(LEXER)>
   recover(mse);
   throw mse;
<else>
   throw mse;
   <! use following code to make it recover inline; remove throw mse;
   recoverFromMismatchedSet(input,mse,FOLLOW_set_in_<ruleName><elementIndex>);
   !>
<endif>
}<\n>
>>

matchRuleBlockSet ::= matchSet
```

```
matchSetAndListLabel(s,label,elementIndex,postmatchCode) ::= <<
<matchSet(...)>
<listLabel(elem=label,...)>
>>

/** Match a string literal */
lexerStringRef(string,label) ::= <<
<if(label)>
int <label>Start = getCharIndex();
match(<string>); <checkRuleBacktrackFailure()>
<label> = new CommonToken(input,
 Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL, <label>Start, getCharIndex()-1);
<else>
match(<string>); <checkRuleBacktrackFailure()><\n>
<endif>
>>

wildcard(label,elementIndex) ::= <<
<if(label)>
<label>=(<labelType>)input.LT(1);<\n>
<endif>
matchAny(input); <checkRuleBacktrackFailure()>
>>

wildcardAndListLabel(label,elementIndex) ::= <<
<wildcard(...)>
<listLabel(elem=label,...)>
>>

/** Match . wildcard in lexer */
wildcardChar(label, elementIndex) ::= <<
<if(label)>
<label> = input.LA(1);<\n>
<endif>
matchAny(); <checkRuleBacktrackFailure()>
>>

wildcardCharListLabel(label, elementIndex) ::= <<
<wildcardChar(...)>
<listLabel(elem=label,...)>
>>

/** Match a rule reference by invoking it possibly with arguments
*  and a return value or values.  The 'rule' argument was the
*  target rule name, but now is type Rule, whose toString is
*  same: the rule name.  Now though you can access full rule
*  descriptor stuff.
*/
```

```
ruleRef(rule,label,elementIndex,args,scope) ::= <<
pushFollow(FOLLOW_<rule.name>_in_<ruleName><elementIndex>);
<if(label)><label>=<endif><if(scope)><scope:delegateName()>.<endif><rule.name>(<args;
 separator=", ">);<\n>
state._fsp--;
<checkRuleBacktrackFailure()>
>>

/** ids+=r */
ruleRefAndListLabel(rule,label,elementIndex,args,scope) ::= <<
<ruleRef(...)>
<listLabel(elem=label,...)>
>>

/** A lexer rule reference.
*
*  The 'rule' argument was the target rule name, but now
*  is type Rule, whose toString is same: the rule name.
*  Now though you can access full rule descriptor stuff.
*/
lexerRuleRef(rule,label,args,elementIndex,scope) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<label> = new CommonToken(input, Token.INVALID_TOKEN_TYPE, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
<if(scope)><scope:delegateName()>.<endif>m<rule.name>(<args; separator=", ">);
<checkRuleBacktrackFailure()>
<endif>
>>

/**
 i+=INT in lexer */
lexerRuleRefAndListLabel(rule,label,args,elementIndex,scope) ::= <<
<lexerRuleRef(...)>
<listLabel(elem=label,...)>
>>

/** EOF in the lexer */
lexerMatchEOF(label,elementIndex) ::= <<
<if(label)>
int <label>Start<elementIndex> = getCharIndex();
match(EOF); <checkRuleBacktrackFailure()>
<labelType> <label> = new CommonToken(input, EOF, Token.DEFAULT_CHANNEL,
<label>Start<elementIndex>, getCharIndex()-1);
<else>
```

```
match(EOF); <checkRuleBacktrackFailure()>
<endif>
>>

/** match ^(root children) in tree parser */
tree(root, actionsAfterRoot, children, nullableChildList,
    enclosingTreeLevel, treeLevel) ::= <<
<root:element()>
<actionsAfterRoot:element()>
<if(nullableChildList)>
if ( input.LA(1)==Token.DOWN ) {
   match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
   <children:element()>
   match(input, Token.UP, null); <checkRuleBacktrackFailure()>
}
<else>
match(input, Token.DOWN, null); <checkRuleBacktrackFailure()>
<children:element()>
match(input,
 Token.UP, null); <checkRuleBacktrackFailure()>
<endif>
>>

/** Every predicate is used as a validating predicate (even when it is
*  also hoisted into a prediction expression).
*/
validateSemanticPredicate(pred,description) ::= <<
if ( !(<evalPredicate(...)>) ) {
   <ruleBacktrackFailure()>
   throw new FailedPredicateException(input, "<ruleName>", "<description>");
}
>>

// F i x e d  D F A  (if-then-else)

dfaState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
else {
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   NoViableAltException nvae =
       new NoViableAltException("<description>", <decisionNumber>, <stateNumber>, input);<\n>
   <@noViableAltException()>
   throw nvae;<\n>
<endif>
}
>>

/** Same as a normal DFA state except that we don't examine lookahead
*  for the bypass alternative.
  It delays error detection but this
*  is faster, smaller, and more what people expect.  For (X)? people
*  expect "if ( LA(1)==X ) match(X);" and that's it.
*/
dfaOptionalBlockState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse ">
>>

/** A DFA state that is actually the loopback decision of a closure
*  loop.  If end-of-token (EOT) predicts any of the targets then it
*  should act like a default clause (i.e., no error can be generated).
*  This is used only in the lexer so that for ('a')* on the end of a rule
*  anything other than 'a' predicts exiting.
*/
dfaLoopbackState(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(<k>);<\n>
<edges; separator="\nelse "><\n>
<if(eotPredictsAlt)>
<if(!edges)>
alt<decisionNumber>=<eotPredictsAlt>; <! if no edges, don't gen ELSE !>
<else>
else {
   alt<decisionNumber>=<eotPredictsAlt>;
}<\n>
<endif>
<endif>
>>

/**
 An accept state indicates a unique alternative has been predicted */
dfaAcceptState(alt) ::= "alt<decisionNumber>=<alt>;"

/** A simple edge with an expression.  If the expression is satisfied,
*  enter to the target state.  To handle gated productions, we may
*  have to evaluate some predicates for this edge.
*/
dfaEdge(labelExpr, targetState, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {
   <targetState>
}
>>
```

```
// Fixed DFA (switch case)

/** A DFA state where a SWITCH may be generated.  The code generator
*  decides if this is possible: CodeGenerator.canGenerateSwitch().
*/
dfaStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n">
default:
<if(eotPredictsAlt)>
   alt<decisionNumber>=<eotPredictsAlt>;
<else>
   <ruleBacktrackFailure()>
   NoViableAltException nvae =
       new NoViableAltException("<description>",
 <decisionNumber>, <stateNumber>, input);<\n>
   <@noViableAltException()>
   throw nvae;<\n>
<endif>
}<\n>
>>

dfaOptionalBlockStateSwitch(k,edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
   <edges; separator="\n">
}<\n>
>>

dfaLoopbackStateSwitch(k, edges,eotPredictsAlt,description,stateNumber,semPredState) ::= <<
switch ( input.LA(<k>) ) {
<edges; separator="\n"><\n>
<if(eotPredictsAlt)>
default:
   alt<decisionNumber>=<eotPredictsAlt>;
   break;<\n>
<endif>
}<\n>
>>

dfaEdgeSwitch(labels, targetState) ::= <<
<labels:{case <it>:}; separator="\n">
   {
   <targetState>
   }
   break;
>>

// Cyclic DFA
```

```
/** The code to initiate execution of a cyclic DFA; this is used
*  in the rule to predict an alt just like the fixed DFA case.
*  The <name> attribute is inherited via the parser, lexer, ...
*/
dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = dfa<decisionNumber>.predict(input);
>>

/* Dump
 DFA tables as run-length-encoded Strings of octal values.
* Can't use hex as compiler translates them before compilation.
* These strings are split into multiple, concatenated strings.
* Java puts them back together at compile time thankfully.
* Java cannot handle large static arrays, so we're stuck with this
* encode/decode approach.  See analysis and runtime DFA for
* the encoding methods.
*/
cyclicDFA(dfa) ::= <<
static final String DFA<dfa.decisionNumber>_eotS =
   "<dfa.javaCompressedEOT; wrap="\"+\n    \"">";
static final String DFA<dfa.decisionNumber>_eofS =
   "<dfa.javaCompressedEOF; wrap="\"+\n    \"">";
static final String DFA<dfa.decisionNumber>_minS =
   "<dfa.javaCompressedMin; wrap="\"+\n    \"">";
static final String DFA<dfa.decisionNumber>_maxS =
   "<dfa.javaCompressedMax; wrap="\"+\n    \"">";
static final String DFA<dfa.decisionNumber>_acceptS =
   "<dfa.javaCompressedAccept; wrap="\"+\n    \"">";
static final String DFA<dfa.decisionNumber>_specialS
 =
   "<dfa.javaCompressedSpecial; wrap="\"+\n    \"">}>";
static final String[] DFA<dfa.decisionNumber>_transitionS = {
       <dfa.javaCompressedTransition:{s|"<s; wrap="\"+\n\"">"}; separator=",\n">
};

static final short[] DFA<dfa.decisionNumber>_eot =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eotS);
static final short[] DFA<dfa.decisionNumber>_eof =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_eofS);
static final char[] DFA<dfa.decisionNumber>_min =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_minS);
static final char[] DFA<dfa.decisionNumber>_max =
DFA.unpackEncodedStringToUnsignedChars(DFA<dfa.decisionNumber>_maxS);
static final short[] DFA<dfa.decisionNumber>_accept =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_acceptS);
static final short[] DFA<dfa.decisionNumber>_special =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_specialS);
```

```
static final short[][] DFA<dfa.decisionNumber>_transition;

static {
  int numStates = DFA<dfa.decisionNumber>_transitionS.length;
   DFA<dfa.decisionNumber>_transition = new short[numStates][];
  for (int i=0; i\<numStates; i++) {
      DFA<dfa.decisionNumber>_transition[i] =
DFA.unpackEncodedString(DFA<dfa.decisionNumber>_transitionS[i]);
  }
}

class DFA<dfa.decisionNumber> extends DFA {

  public DFA<dfa.decisionNumber>(BaseRecognizer recognizer) {
      this.recognizer = recognizer;
      this.decisionNumber = <dfa.decisionNumber>;
      this.eot = DFA<dfa.decisionNumber>_eot;
      this.eof = DFA<dfa.decisionNumber>_eof;
      this.min = DFA<dfa.decisionNumber>_min;
      this.max = DFA<dfa.decisionNumber>_max;
      this.accept = DFA<dfa.decisionNumber>_accept;
      this.special = DFA<dfa.decisionNumber>_special;
      this.transition = DFA<dfa.decisionNumber>_transition;
  }
  public String getDescription() {
      return "<dfa.description>";
  }
  <@errorMethod()>
<if(dfa.specialStateSTs)>
  public int specialStateTransition(int s, IntStream _input)
 throws NoViableAltException {
      <if(LEXER)>
      IntStream input = _input;
      <endif>
      <if(PARSER)>
      TokenStream input = (TokenStream)_input;
      <endif>
      <if(TREE_PARSER)>
      TreeNodeStream input = (TreeNodeStream)_input;
      <endif>
   int _s = s;
      switch ( s ) {
      <dfa.specialStateSTs:{state |
      case <i0> : <! compressed special state numbers 0..n-1 !>
          <state>}; separator="\n">
       }
<if(backtracking)>
      if (state.backtracking>0) {state.failed=true; return -1;}<\n>
<endif>
    NoViableAltException nvae =
      new NoViableAltException(getDescription(), <dfa.decisionNumber>, _s, input);
    error(nvae);
    throw nvae;
  }<\n>
<endif>
}<\n>
>>

/** A state in a cyclic DFA; it's a special state and part of a big switch on
*  state.
*/
cyclicDFAState(decisionNumber,stateNumber,edges,needErrorClause,semPredState) ::= <<
int LA<decisionNumber>_<stateNumber> = input.LA(1);<\n>
<if(semPredState)>
 <! get next lookahead symbol to test edges, then rewind !>
int index<decisionNumber>_<stateNumber> = input.index();
input.rewind();<\n>
<endif>
s = -1;
<edges; separator="\nelse ">
<if(semPredState)> <! return input cursor to state before we rewound !>
input.seek(index<decisionNumber>_<stateNumber>);<\n>
<endif>
if ( s>=0 ) return s;
break;
>>

/** Just like a fixed DFA edge, test the lookahead and indicate what
*  state to jump to next if successful.
*/
cyclicDFAEdge(labelExpr, targetStateNumber, edgeNumber, predicates) ::= <<
if ( (<labelExpr>) <if(predicates)>&& (<predicates>)<endif>) {s = <targetStateNumber>;}<\n>
>>

/** An edge pointing at end-of-token; essentially matches any char;
*  always jump to the target.
*/
eotDFAEdge(targetStateNumber,edgeNumber, predicates) ::= <<
s = <targetStateNumber>;<\n>
>>


// D F A  E X P R E S S I O N S

andPredicates(left,right) ::= "(<left>&&<right>)"
```

```
orPredicates(operands) ::= "(<first(operands)><rest(operands):{o |
 ||<o>}>)"

notPredicate(pred) ::= "!(<evalPredicate(...)>)"

evalPredicate(pred,description) ::= "(<pred>)"

evalSynPredicate(pred,description) ::= "<pred>()"

lookaheadTest(atom,k,atomAsInt) ::= "LA<decisionNumber>_<stateNumber>==<atom>"

/** Sometimes a lookahead test cannot assume that LA(k) is in a temp variable
*  somewhere.  Must ask for the lookahead directly.
*/
isolatedLookaheadTest(atom,k,atomAsInt) ::= "input.LA(<k>)==<atom>"

lookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= <<
(LA<decisionNumber>_<stateNumber>\>=<lower> && LA<decisionNumber>_<stateNumber>\<=<upper>)
>>

isolatedLookaheadRangeTest(lower,upper,k,rangeNumber,lowerAsInt,upperAsInt) ::= "(input.LA(<k>)\>=<lower>
&& input.LA(<k>)\<=<upper>)"

setTest(ranges) ::= "<ranges; separator=\"||\">"

// A T T R I B U T E S

globalAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack
 = new Stack();<\n>
<endif>
>>

ruleAttributeScope(scope) ::= <<
<if(scope.attributes)>
protected static class <scope.name>_scope {
   <scope.attributes:{<it.decl>;}; separator="\n">
}
protected Stack <scope.name>_stack = new Stack();<\n>
<endif>
>>

returnStructName() ::= "<it.name>_return"
```

```
returnType() ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
<ruleDescriptor.grammar.recognizerName>.<ruleDescriptor:returnStructName()>
<else>
<if(ruleDescriptor.hasSingleReturnValue)>
<ruleDescriptor.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

/** Generate the Java type associated with a single or multiple return
*  values.
*/
ruleLabelType(referencedRule) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
<referencedRule.grammar.recognizerName>.<referencedRule.name>_return
<else>
<if(referencedRule.hasSingleReturnValue)>
<referencedRule.singleValueReturnType>
<else>
void
<endif>
<endif>
>>

delegateName() ::= <<
<if(it.label)><it.label><else>g<it.name><endif>
>>

/**
 Using a type to init value map, try to init a type; if not in table
*  must be an object, default value is "null".
*/
initValue(typeName) ::= <<
<javaTypeInitMap.(typeName)>
>>

/** Define a rule label including default value */
ruleLabelDef(label) ::= <<
<ruleLabelType(referencedRule=label.referencedRule)> <label.label.text> =
<initValue(typeName=ruleLabelType(referencedRule=label.referencedRule))>;<\n>
>>

/** Define a return struct for a rule if the code needs to access its
*  start/stop tokens, tree stuff, attributes, ...  Leave a hole for
*  subgroups to stick in members.
```

```
*/
returnScope(scope) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
public static class <ruleDescriptor:returnStructName()> extends
<if(TREE_PARSER)>Tree<else>Parser<endif>RuleReturnScope {
   <scope.attributes:{public <it.decl>;}; separator="\n">
   <@ruleReturnMembers()>
};
<endif>
>>

parameterScope(scope) ::= <<
<scope.attributes:{<it.decl>}; separator=", ">
>>

parameterAttributeRef(attr)
 ::= "<attr.name>"
parameterSetAttributeRef(attr,expr) ::= "<attr.name> =<expr>;"

scopeAttributeRef(scope,attr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name>
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name>
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name>
<endif>
<endif>
>>

scopeSetAttributeRef(scope,attr,expr,index,negIndex) ::= <<
<if(negIndex)>
((<scope>_scope)<scope>_stack.elementAt(<scope>_stack.size()-<negIndex>-1)).<attr.name> =<expr>;
<else>
<if(index)>
((<scope>_scope)<scope>_stack.elementAt(<index>)).<attr.name> =<expr>;
<else>
((<scope>_scope)<scope>_stack.peek()).<attr.name> =<expr>;
<endif>
<endif>
>>

/** $x is either global scope or x is rule with dynamic scope; refers
*  to stack itself not top of stack.  This is useful for predicates
*  like {$function.size()>0 && $function::name.equals("foo")}?
*/
isolatedDynamicScopeRef(scope) ::= "<scope>_stack"
```

```
/**
 reference an attribute of rule; might only have single return value */
ruleLabelRef(referencedRule,scope,attr) ::= <<
<if(referencedRule.hasMultipleReturnValues)>
(<scope>!=null?<scope>.<attr.name>:<initValue(attr.type)>)
<else>
<scope>
<endif>
>>

returnAttributeRef(ruleDescriptor,attr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name>
<else>
<attr.name>
<endif>
>>

returnSetAttributeRef(ruleDescriptor,attr,expr) ::= <<
<if(ruleDescriptor.hasMultipleReturnValues)>
retval.<attr.name> =<expr>;
<else>
<attr.name> =<expr>;
<endif>
>>

/** How to translate $tokenLabel */
tokenLabelRef(label) ::= "<label>"

/** ids+=ID {$ids} or e+=expr {$e} */
listLabelRef(label) ::= "list_<label>"



// not sure the next are the right approach

tokenLabelPropertyRef_text(scope,attr) ::= "(<scope>!=null?<scope>.getText():null)"
tokenLabelPropertyRef_type(scope,attr) ::= "(<scope>!=null?<scope>.getType():0)"
tokenLabelPropertyRef_line(scope,attr) ::= "(<scope>!=null?<scope>.getLine():0)"
tokenLabelPropertyRef_pos(scope,attr)
 ::= "(<scope>!=null?<scope>.getCharPositionInLine():0)"
tokenLabelPropertyRef_channel(scope,attr) ::= "(<scope>!=null?<scope>.getChannel():0)"
tokenLabelPropertyRef_index(scope,attr) ::= "(<scope>!=null?<scope>.getTokenIndex():0)"
tokenLabelPropertyRef_tree(scope,attr) ::= "<scope>_tree"
tokenLabelPropertyRef_int(scope,attr) ::= "(<scope>!=null?Integer.valueOf(<scope>.getText()):0)"

ruleLabelPropertyRef_start(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.start):null)"
ruleLabelPropertyRef_stop(scope,attr) ::= "(<scope>!=null?((<labelType>)<scope>.stop):null)"
```

```
ruleLabelPropertyRef_tree(scope,attr) ::= "(<scope>!=null?((<ASTLabelType>)<scope>.tree):null)"
ruleLabelPropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
(<scope>!=null?(input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(<scope>.start),
 input.getTreeAdaptor().getTokenStopIndex(<scope>.start))):null)
<else>
(<scope>!=null?input.toString(<scope>.start,<scope>.stop):null)
<endif>
>>

ruleLabelPropertyRef_st(scope,attr)
 ::= "(<scope>!=null?<scope>.st:null)"

/** Isolated $RULE ref ok in lexer as it's a Token */
lexerRuleLabel(label) ::= "<label>"

lexerRuleLabelPropertyRef_type(scope,attr) ::=
   "(<scope>!=null?<scope>.getType():0)"
lexerRuleLabelPropertyRef_line(scope,attr) ::=
   "(<scope>!=null?<scope>.getLine():0)"
lexerRuleLabelPropertyRef_pos(scope,attr) ::=
   "(<scope>!=null?<scope>.getCharPositionInLine():-1)"
lexerRuleLabelPropertyRef_channel(scope,attr) ::=
   "(<scope>!=null?<scope>.getChannel():0)"
lexerRuleLabelPropertyRef_index(scope,attr) ::=
   "(<scope>!=null?<scope>.getTokenIndex():0)"
lexerRuleLabelPropertyRef_text(scope,attr) ::=
   "(<scope>!=null?<scope>.getText():null)"
lexerRuleLabelPropertyRef_int(scope,attr) ::=
   "(<scope>!=null?Integer.valueOf(<scope>.getText()):0)"

// Somebody may ref $template or $tree or $stop within a rule:
rulePropertyRef_start(scope,attr) ::= "((<labelType>)retval.start)"
rulePropertyRef_stop(scope,attr)
 ::= "((<labelType>)retval.stop)"
rulePropertyRef_tree(scope,attr) ::= "((<ASTLabelType>)retval.tree)"
rulePropertyRef_text(scope,attr) ::= <<
<if(TREE_PARSER)>
input.getTokenStream().toString(
 input.getTreeAdaptor().getTokenStartIndex(retval.start),
 input.getTreeAdaptor().getTokenStopIndex(retval.start))
<else>
input.toString(retval.start,input.LT(-1))
<endif>
>>
rulePropertyRef_st(scope,attr) ::= "retval.st"
```

```
lexerRulePropertyRef_text(scope,attr) ::= "getText()"
lexerRulePropertyRef_type(scope,attr) ::= "_type"
lexerRulePropertyRef_line(scope,attr) ::= "state.tokenStartLine"
lexerRulePropertyRef_pos(scope,attr) ::= "state.tokenStartCharPositionInLine"
lexerRulePropertyRef_index(scope,attr) ::= "-1" // undefined token index in lexer
lexerRulePropertyRef_channel(scope,attr) ::= "_channel"
lexerRulePropertyRef_start(scope,attr) ::= "state.tokenStartCharIndex"
lexerRulePropertyRef_stop(scope,attr) ::= "(getCharIndex()-1)"
lexerRulePropertyRef_int(scope,attr)
 ::= "Integer.valueOf(<scope>.getText())"

// setting $st and $tree is allowed in local rule. everything else
// is flagged as error
ruleSetPropertyRef_tree(scope,attr,expr) ::= "retval.tree =<expr>;"
ruleSetPropertyRef_st(scope,attr,expr) ::= "retval.st =<expr>;"

/** How to execute an action (only when not backtracking) */
execAction(action) ::= <<
<if(backtracking)>
if ( <actions.(actionScope).synpredgate> ) {
 <action>
}
<else>
<action>
<endif>
>>

/** How to always execute an action even when backtracking */
execForcedAction(action) ::= "<action>"

// M I S C (properties, etc...)

bitset(name, words64) ::= <<
public static final BitSet <name> = new BitSet(new long[]{<words64:{<it>L};separator=",">});<\n>
>>

codeFileExtension() ::= ".java"

true() ::= "true"
false() ::= "false"
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/Java/Java.stg
No license file was found, but licenses were detected in source scan.

/*
[The "BSD licence"]

Copyright (c) 2005-2009 Jim Idle, Temporal Wave LLC
http://www.temporal-wave.com
http://www.linkedin.com/in/jimidle

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
/** Template overrides to add debugging to normal C output;
*  If ASTs are built, then you'll also get ASTDbg.stg loaded.
*/
group Dbg;

@genericParser.members() ::= <<
<if(grammar.grammarIsRoot)>
const char *
ruleNames[] =
 {
  "invalidRule", <grammar.allImportedRules:{rST | "<rST.name>"}; wrap="\n  ", separator=", ">
 };<\n>
<endif>
<if(grammar.grammarIsRoot)> <! grammar imports other grammar(s)
 !>
static ANTLR3_UINT32 ruleLevel = 0;
static ANTLR3_UINT32 getRuleLevel()
{
 return ruleLevel;
}
static void incRuleLevel()
{
 ruleLevel++;
}
static void decRuleLevel()
{
 ruleLevel--;
}
<else> <! imported grammar !>
static ANTLR3_UINT32
getRuleLevel()
{
 return <grammar.delegators:{g| <g:delegateName()>}>->getRuleLevel();
}
static void incRuleLevel()
{
 <grammar.delegators:{g| <g:delegateName()>}>->incRuleLevel();
}
static void
decRuleLevel()
{
 <grammar.delegators:{g| <g:delegateName()>}>.decRuleLevel();
}
<endif>
<if(profile)>
// Profiling not yet implemented for C target
//
<endif>
<if(grammar.grammarIsRoot)>
<ctorForPredefinedListener()>
<else>
<ctorForDelegateGrammar()>
<endif>

static ANTLR3_BOOLEAN
evalPredicate(p<name> ctx, ANTLR3_BOOLEAN result, const char * predicate)
{
   DBG->semanticPredicate(DBG, result, predicate);
   return result;
}<\n>
>>

@genericParser.debugStuff() ::= <<
<if(grammar.grammarIsRoot)>
<createListenerAndHandshake()>
<endif>
>>
```

```
ctorForProfilingRootGrammar()
 ::= <<
>>

/** Basically we don't want to set any dbg listeners as root will have it. */
ctorForDelegateGrammar() ::= <<

>>

ctorForPredefinedListener() ::= <<

>>

createListenerAndHandshake() ::= <<
{
 // DEBUG MODE code
 //
 pANTLR3_DEBUG_EVENT_LISTENER  proxy;
 proxy = antlr3DebugListenerNew();
 proxy->grammarFileName = INPUT->tokenSource->strFactory->newStr8(INPUT->tokenSource->strFactory,
(pANTLR3_UINT8)ctx->getGrammarFileName());

<if(TREE_PARSER)>
 proxy->adaptor = ADAPTOR;
<endif>
 PARSER->setDebugListener(PARSER, proxy);

 // Try to connect to the debugger (waits forever for a connection)
 //
 proxy->handshake(proxy);

 // End DEBUG MODE code
 //
}
>>



@rule.preamble() ::= <<
DBG->enterRule(DBG, getGrammarFileName(), (const char *)"<ruleName>");
if ( getRuleLevel()==0 )
{
 DBG->commence(DBG);
}
incRuleLevel();
DBG->location(DBG, <ruleDescriptor.tree.line>, <ruleDescriptor.tree.column>);<\n>
>>
```

```
@rule.postamble() ::= <<
DBG->location(DBG,
 <ruleDescriptor.EORNode.line>, <ruleDescriptor.EORNode.column>);<\n>
DBG->exitRule(DBG, getGrammarFileName(), (const char *)"<ruleName>");
decRuleLevel();
if ( getRuleLevel()==0 )
{
 DBG->terminate(DBG);
}
<\n>
>>

@synpred.start() ::= "DBG->beginBacktrack(DBG, BACKTRACKING);"

@synpred.stop() ::= "DBG->endBacktrack(DBG, BACKTRACKING, success);"

// Common debug event triggers used by region overrides below

enterSubRule() ::=
   "DBG->enterSubRule(DBG, <decisionNumber>);<\n>"

exitSubRule() ::=
   "DBG->exitSubRule(DBG, <decisionNumber>);<\n>"

enterDecision() ::=
   "DBG->enterDecision(DBG, <decisionNumber>);<\n>"

exitDecision() ::=
   "DBG->exitDecision(DBG, <decisionNumber>);<\n>"

enterAlt(n) ::= "DBG->enterAlt(DBG, <n>);<\n>"

// Region overrides that tell various constructs to add debugging triggers

@block.predecision() ::= "<enterSubRule()><enterDecision()>"

@block.postdecision() ::= "<exitDecision()>"

@block.postbranch() ::= "<exitSubRule()>"

@ruleBlock.predecision()
 ::= "<enterDecision()>"

@ruleBlock.postdecision() ::= "<exitDecision()>"

@ruleBlockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"

@blockSingleAlt.prealt() ::= "<enterAlt(n=\"1\")>"
```

```
@positiveClosureBlock.preloop() ::= "<enterSubRule()>"

@positiveClosureBlock.postloop() ::= "<exitSubRule()>"

@positiveClosureBlock.predecision() ::= "<enterDecision()>"

@positiveClosureBlock.postdecision() ::= "<exitDecision()>"

@positiveClosureBlock.earlyExitException() ::=
   "DBG->recognitionException(DBG, EXCEPTION);<\n>"

@closureBlock.preloop() ::= "<enterSubRule()>"

@closureBlock.postloop() ::= "<exitSubRule()>"

@closureBlock.predecision() ::= "<enterDecision()>"

@closureBlock.postdecision() ::= "<exitDecision()>"

@altSwitchCase.prealt() ::= "<enterAlt(n=i)>"

@element.prematch() ::=
   "DBG->location(DBG, <it.line>, <it.pos>);"

@matchSet.mismatchedSetException() ::=
   "DBG->recognitionException(DBG, EXCEPTION);"

@newNVException.noViableAltException() ::= "DBG->recognitionException(DBG,
 EXCEPTION);"

dfaDecision(decisionNumber,description) ::= <<
alt<decisionNumber> = cdfa<decisionNumber>.predict(ctx, RECOGNIZER, ISTREAM,
&cdfa<decisionNumber>);
if  (HASEXCEPTION())
{
 DBG->recognitionException(DBG, EXCEPTION);
   goto rule<ruleDescriptor.name>Ex;
}
<checkRuleBacktrackFailure()>
>>

@cyclicDFA.errorMethod() ::= <<
//static void
//dfaError(p<name> ctx)
//{
//    DBG->recognitionException(DBG, EXCEPTION);
//}
```

```
>>

/** Force predicate validation to trigger an event */
evalPredicate(pred,description) ::= <<
evalPredicate(ctx, <pred>, (const char *)"<description>")
>>
```

Found in path(s):
* /opt/cola/permits/175342058_1695326769.3349802/0/antlr-3-1-3-sources-2-jar/org/antlr/codegen/templates/C/Dbg.stg

1.86 avro-guava-dependencies 1.8.2

1.86.1 Available under license :

Apache Avro Guava Dependencies
Copyright 2009-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

License for Guava classes included in this binary artifact:

Copyright: 2006-2015 The Guava Authors
License: http://www.apache.org/licenses/LICENSE-2.0 (see above)

1.87 aop-alliance 1.0.0

1.88 grizzly-framework 2.3.25

1.88.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
 subject to such option by the copyright
* holder.
*/
```

Found in path(s):

```
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/Holder.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2011 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
```

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/DefaultProcessorSelector.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Transformer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/MemoryUtils.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/FilterChainProcessorSelector.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ChainProcessorSelector.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/ExecutorResolver.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/TransformationResult.java
*
/opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ProcessorResult.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/TransportFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/DefaultSelectionKeyHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/LoggingFormatter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/AbstractStreamReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/RegisterChannelResult.java

No license file was found, but licenses were detected in source scan.

* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/Encoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/NIOTransportBuilder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/DataStructures.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/LZMAFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/Futures.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/GenericAdapter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/LZMADecoder.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/rangecoder/RangeEncoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/Base.java

* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/BufferInputStream.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/Decoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/LZMAEncoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/rangecoder/BitTreeDecoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/rangecoder/BitTreeEncoder.java
*
/opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ReadHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/PushBackHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/lz/BinTree.java
No license file was found, but licenses were detected in source scan.

* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/licenses/publicdomain
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/TransferQueue.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields

* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ConcurrentHashMapV8.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ThreadLocalRandom.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"

* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/RecordReadResult.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/RecordWriteResult.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/GracefulShutdownRunner.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/CloseReason.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*

* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/NIOConnection.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/NullaryFunction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/Filter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLDecoderTransformer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/IOEvent.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Transport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/tmpselectors/TemporarySelectorsEnabledTransport.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/NextAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/ListFacadeFilterChain.java

* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/Buffers.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Reader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/AttributeStorage.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/BaseFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ObjectPool.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/conditions/Condition.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncQueueWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Context.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Result.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Codec.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ConnectorHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/SelectorHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/AbstractNIOConnectionDistributor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/NullaryFunction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/FilterExecutor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/tmpselectors/TemporarySelectorIO.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/StreamReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncQueue.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncQueueEnabledTransport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ProcessorSelector.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Writer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/AttributeHolder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-

jar/org/glassfish/grizzly/filterchain/FilterChainContext.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/MemoryManager.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/PoolableObject.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/Attribute.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/StreamWriter.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/AttributeBuilder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOServerConnection.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/impl/FutureImpl.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Buffer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/WorkerThread.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/AbstractReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Readable.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLUtils.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ExceptionHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/MessageCloner.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOConnection.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/WriteResult.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/FilterChainEnabledTransport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/FilterChainBuilder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOTransport.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/IndexedAttributeHolder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/AbstractFilterChain.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/tmpselectors/TemporarySelectorWriter.java

* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ReadResult.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/TaskQueue.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/NIOChannelDistributor.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/SelectionKeyHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/SocketAcceptor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/IndexedAttributeAccessor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/SelectorRunner.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLEngineConfigurator.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncQueueIO.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/DefaultFilterChain.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLEncoderTransformer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ByteBufferWrapper.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Connection.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/DefaultSelectorHandler.java
No license file was found, but licenses were detected in source scan.

* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/DefaultStreamWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOConnectorHandler.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2007-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLContextConfigurator.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/MemoryManagerInitializer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOBindingHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/InputBufferWrapper.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/AbstractBindingHandler.java
No license file was found, but licenses were detected in source scan.

* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/LinkedTransferQueue.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2016 Oracle and/or its affiliates. All rights reserved.
*

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLConnectionContext.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOUtils.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/DirectByteBufferRecord.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/Exceptions.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/BufferOutputStream.java
No license file was found, but licenses were detected in source scan.

/*

* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2009-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLSupport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLSupportImpl.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLEngineFactory.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/RerunFilterAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/BufferedInput.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/StreamInput.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/TransformerStreamReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ServiceConfigurationError.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/TransformerStreamWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/StringFilter.java
No license file was found, but licenses were detected in source scan.

 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/PushBackContext.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2009-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/strategies/WorkerThreadIOStrategy.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/IOStrategy.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncQueueRecord.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/strategies/SameThreadIOStrategy.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/strategies/SimpleDynamicNIOStrategy.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/strategies/LeaderFollowerNIOStrategy.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/zip/GZipDecoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/PendingWriteQueueLimitExceededException.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/AbstractBufferArray.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ArraySet.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/impl/UnsafeFutureImpl.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/FixedThreadPool.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/monitoring/DefaultMonitoringConfig.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/jmxbase/GrizzlyJmxManager.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ByteBufferArray.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-

jar/org/glassfish/grizzly/threadpool/ThreadPoolConfig.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/GrizzlyExecutorService.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/strategies/AbstractIOStrategy.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/zip/GZipEncoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/InputSource.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/DebugPoint.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/OutputSink.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/BufferArray.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ThreadCache.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ProbeNotifier.java
No license file was found, but licenses were detected in source scan.

* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOTransportBuilder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOTransportBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ProcessorExecutor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/AbstractNIOAsyncQueueReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/UnsafeAttributeHolder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/FilterChain.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/RoundRobinConnectionDistributor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOAsyncQueueWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/AbstractTransformer.java
*
/opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOTransport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/AbstractNIOAsyncQueueWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOAsyncQueueWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/AbstractStreamWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncReadQueueRecord.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/NIOTransport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncWriteQueueRecord.java

*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ByteBufferManager.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOConnectorHandler.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/CloseType.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/GracefulShutdownListener.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/Threads.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/monitoring/MonitoringUtils.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ShutdownContext.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ICloseType.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/CloseListener.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/GenericCloseListener.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLBaseFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/JdkVersion.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOBindingHandler.java
No license file was found, but licenses were detected in source scan.

* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/DelayFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/CompletionHandlerAdapter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/StreamDecodeCondition.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ThreadLocalPoolProvider.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ServiceFinder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/Bits.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ThreadLocalPool.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ArrayUtils.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/localization/LocalizableMessageFactory.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/localization/Localizable.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/strategies/WorkerThreadPoolConfigProducer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/monitoring/MonitoringAware.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/ForkAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/localization/Localizer.java
*

/opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/localization/LocalizableMessage.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-

jar/org/glassfish/grizzly/compression/lzma/impl/CRC.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/lz/OutWindow.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ChunkingFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/lz/InWindow.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/WritableMessage.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/FileTransfer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/WriteHandler.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/lzma/impl/rangecoder/RangeDecoder.java
No license file was found, but licenses were detected in source scan.

* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ActivityCheckFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOTemporarySelectorWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOTransportFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOTemporarySelectorReader.java
No license file was found, but licenses were detected in source scan.

# Copyright (c) 2009-2010 Oracle and/or its affiliates. All rights reserved.
# The contents of this file are subject to the terms of either the GNU
# General Public License Version 2 only ("GPL") or the Common Development
# and Distribution License("CDDL") (collectively, the "License"). You
# may not use this file except in compliance with the License. You can
# Oracle designates this particular file as subject to the "Classpath"
# exception as provided by Oracle in the GPL Version 2 section of the License

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/version.properties
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific

* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/TransformerOutput.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/impl/SafeFutureImpl.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/TransformerInput.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ReadOnlyHeapBuffer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/QueueLimitedThreadPool.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/CompositeBuffer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/HeapMemoryManager.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/BufferedOutput.java

No license file was found, but licenses were detected in source scan.

DO NOT TRANSLATE OR LOCALIZE.

===
- TABLE OF CONTENTS -
===

I. Components

===
I. COMPONENTS
===

The following software (or certain identified files distributed with the software) may be included in this product. Unless otherwise specified, the software identified in this file is licensed under the licenses described below. The disclaimers and copyright notices provided are based on information made available to Oracle by the third party licensors listed.

****

Dojo 1.2.3

The "New" BSD License:

****

Copyright (c) 2005-2010, The Dojo Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Dojo Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

32 ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
33 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
34 DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 CONTRIBUTORS BE LIABLE
35 FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
36 DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
37 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
38 CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
39 OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
40 OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LinkedTransferQueue.java

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below
 (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed,
 transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/Grizzly_THIRDPARTYLICENSEREADME.txt
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2015 Oracle and/or its affiliates. All rights reserved.
*

* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/PooledMemoryManager.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Copyable.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2013 Oracle and/or its affiliates. All rights reserved.

*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/SocketConnectorHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/CompletionHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/impl/ReadyFutureImpl.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOTransportFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-

jar/org/glassfish/grizzly/AbstractTransport.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/InternalContextImpl.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/AbstractWriter.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/StandaloneProcessor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Processor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/DefaultWorkerThread.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/TCPNIOAsyncQueueReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOAsyncQueueReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/DefaultAttributeBuilder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Writeable.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/AbstractSocketConnectorHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/tmpselectors/TemporarySelectorReader.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/localization/log.properties
No license file was found, but licenses were detected in source scan.

* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/PortRange.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOTemporarySelectorReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/DefaultStreamReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOTemporarySelectorWriter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLStreamWriter.java
No license file was found, but licenses were detected in source scan.

/*

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/InvokeAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Grizzly.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/StopAction.java

* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/tmpselectors/TemporarySelectorPool.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/AbstractNextAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/StateHolder.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.

*/

Found in path(s):

* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/StreamOutput.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/HeapBuffer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Appender.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/SyncThreadPool.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/Charsets.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/ProbeNotifier.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/compression/zip/GZipFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/IOEventLifeCycleListener.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/MemoryProbe.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ChangeListener.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/SelectorHandlerTask.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/monitoring/MonitoringConfig.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/AbstractCodecFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ConnectionProbe.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Closeable.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/Output.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/ResultAware.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/AbstractMemoryManager.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/GrizzlyFuture.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/ByteBufferAware.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/AbstractThreadPool.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-

jar/org/glassfish/grizzly/Cacheable.java
*
 /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/threadpool/ThreadPoolProbe.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/Input.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Appendable.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/TransportProbe.java
No license file was found, but licenses were detected in source scan.

* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/EchoFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/EmptyCompletionHandler.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/LogFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/Pair.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/RerunChainAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/CodecFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/FilterChainEvent.java
*
/opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/filterchain/SuspendAction.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/asyncqueue/AsyncQueueReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/StandaloneProcessorSelector.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLCodec.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/TransformationException.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2014-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/ChannelConfigurator.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/DefaultMemoryManagerFactory.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOServerConnection.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/StringDecoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/Interceptor.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/transport/UDPNIOConnection.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/BuffersBuffer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/memory/WrapperAware.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/IdleTimeoutFilter.java
*
/opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/SocketBinder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-

jar/org/glassfish/grizzly/utils/SilentConnectionFilter.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/StringEncoder.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/utils/DelayedExecutor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/ssl/SSLStreamReader.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/streams/Stream.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.

*/

Found in path(s):
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/attributes/AttributeBuilderInitializer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/SelectionKeyHandlerInitializer.java
* /opt/cola/permits/175354463_1695324562.0775313/0/grizzly-framework-2-3-25-sources-jar/org/glassfish/grizzly/nio/Selectors.java

1.89 grizzly-http 2.3.25

1.89.1 Available under license :

No license file was found, but licenses were detected in source scan.

* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/KeepAlive.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/ProcessingState.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/GZipContentEncoding.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2007-2011 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/StringManager.java
No license file was found, but licenses were detected in source scan.

* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpBaseFilter.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2007-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/MimeType.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/Base64Utils.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/HeaderValue.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/CompressionConfig.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
```

Found in
 path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/BinaryNIOOutputSink.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpContext.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpBrokenContentException.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/Note.java
No license file was found, but licenses were detected in source scan.

```
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
```

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/FastDateFormat.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/UDecoder.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/B2CConverter.java
*
/opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/MessageBytes.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/UTF8Decoder.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/B2CConverterBlocking.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/InputBuffer.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpEvents.java
No license file was found, but licenses were detected in source scan.

```
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/FastHttpDateFormat.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/CacheableDataChunk.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/ParsingResult.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/ContentEncodingException.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.

*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpRequestPacketImpl.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpResponsePacketImpl.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpPacket.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/Cookies.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/Cookie.java
*
/opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/CookieUtils.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/CookieParserUtils.java
No license file was found, but licenses were detected in source scan.

```
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/TimeStamp.java

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-

jar/org/glassfish/grizzly/http/io/NIOWriter.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpTrailer.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpProbeNotifier.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/NIOInputStream.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/NIOReader.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpContent.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/MimeHeadersPacket.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpResponsePacket.java
*
/opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/CookiesBuilder.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/NIOOutputStream.java
No license file was found, but licenses were detected in source scan.

```
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/CookieSerializerUtils.java
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2011-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
```

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/TemporaryHeapBuffer.java
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
```

* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/C2BConverter.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/HexUtils.java
*
 /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/StringCache.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/LazyCookieState.java
No license file was found, but licenses were detected in source scan.

* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/DataChunk.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpProbe.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/TransferEncoding.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/BufferChunk.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/HttpUtils.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/OutputBuffer.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/Chunk.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpRequestPacket.java
*
/opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/EncodingFilter.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/HttpRequestURIDecoder.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpClientFilter.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpPacketParsing.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/RequestURIRef.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpHeader.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/ChunkedTransferEncoding.java
*
/opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/Protocol.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpCodecFilter.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpServerFilter.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-

jar/org/glassfish/grizzly/http/Method.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/HttpStatus.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/URLDecoder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/ContentType.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/Header.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/HttpCodecUtils.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/ContentEncoding.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/FixedLengthTransferEncoding.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/LZMAContentEncoding.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/io/BinaryNIOInputSource.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/HttpBrokenContent.java
No license file was found, but licenses were detected in source scan.

# Copyright (c) 2010 Oracle and/or its affiliates. All rights reserved.
# The contents of this file are subject to the terms of either the GNU
# General Public License Version 2 only ("GPL") or the Common Development
# and Distribution License("CDDL") (collectively, the "License"). You
# may not use this file except in compliance with the License. You can
# Oracle designates this particular file as subject to the "Classpath"
# exception as provided by Oracle in the GPL Version 2 section of the License

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/version.properties
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/KeepAliveProbe.java
No license file was found, but licenses were detected in source scan.

* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/Constants.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/MimeHeaders.java
*
/opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/Parameters.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/Ascii.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/UEncoder.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-

jar/org/glassfish/grizzly/http/util/ByteChunk.java
* /opt/cola/permits/175354812_1695326807.1014678/0/grizzly-http-2-3-25-sources-jar/org/glassfish/grizzly/http/util/CharChunk.java

1.90 grizzly-http-server 2.3.25

1.90.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2009-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
```

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/AlternateDocBase.java
No license file was found, but licenses were detected in source scan.

as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/GrizzlyPrincipal.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*

* Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/SimpleDateFormats.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/AggregatorAddOn.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/Constants.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2011-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code

* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/filecache/LazyFileCacheKey.java
No license file was found, but licenses were detected in source scan.

* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/CompressionEncodingFilter.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/filecache/FileCacheKey.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/HtmlHelper.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpServerFilter.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/FileCacheFilter.java
No license file was found, but licenses were detected in source scan.

* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/DefaultSessionManager.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpHandlerRegistration.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/AccessLogProbe.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/SessionManager.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpServerProbe.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/ServerFilterConfiguration.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/filecache/FileCache.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/NetworkListener.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/ServerConfiguration.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/filecache/FileCacheProbe.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/Session.java
*
 /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/SuspendStatus.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/SuspendContext.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/io/WriteHandler.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/io/ReadHandler.java
No license file was found, but licenses were detected in source scan.

*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/StaticHttpHandler.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*

* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/DefaultErrorPageGenerator.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each

* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/NIOInputStreamImpl.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/NIOReaderImpl.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/NIOWriterImpl.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/NIOOutputStreamImpl.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at

*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/Enumerator.java
No license file was found, but licenses were detected in source scan.

```
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/Globals.java
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
```

* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/MappingData.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/Mapper.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2007-2011 Oracle and/or its affiliates. All rights reserved.
*

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/ClassLoaderUtil.java
No license file was found, but licenses were detected in source scan.

```
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/StringParser.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/LocaleParser.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/naming/NamingException.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/TagLocaleParser.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/State.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/naming/DirContext.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/LegacyLocaleParser.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/QueueingAppender.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/RotatingFileAppender.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/AccessLogBuilder.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/AccessLogAppender.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/AccessLogFormat.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/StreamAppender.java
*
 /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/HttpPipelineOptAddOn.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/SimpleDateFormatThreadLocal.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/FileAppender.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/accesslog/ApacheLogFormat.java
No license file was found, but licenses were detected in source scan.

```
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/CompressionLevel.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/AddOn.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/RequestUtils.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/DispatcherHelper.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2016 Oracle and/or its affiliates. All rights reserved.
*
```

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpServer.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpHandler.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpHandlerChain.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/TimeoutHandler.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/AfterServiceListener.java
No license file was found, but licenses were detected in source scan.

/*

* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/ParameterMap.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2008-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL

```
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/Request.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/util/ExpandJar.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/Response.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
```

* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpServerProbeNotifier.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/HttpServerMonitoringConfig.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/jmxbase/Monitorable.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/jmxbase/JmxEventListener.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/filecache/FileCacheEntry.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*

* Copyright (c) 2012-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/io/ServerInputBuffer.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/io/ServerOutputBuffer.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/BackendConfiguration.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/ErrorPageGenerator.java

* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/naming/NamingContext.java

* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-

jar/org/glassfish/grizzly/http/server/RequestExecutorProvider.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/StaticHttpHandlerBase.java
* /opt/cola/permits/175355113_1695326812.7110336/0/grizzly-http-server-2-3-25-sources-jar/org/glassfish/grizzly/http/server/CLStaticHttpHandler.java

1.91 jackson-jaxrs-base 2.7.9

1.91.1 Available under license :

1.92 javax-el-api 2.2.4

1.92.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/BeanELResolver.java

No license file was found, but licenses were detected in source scan.

```
# Copyright (c) 1997-2010 Oracle and/or its affiliates. All rights reserved.
# The contents of this file are subject to the terms of either the GNU
# General Public License Version 2 only ("GPL") or the Common Development
# and Distribution License("CDDL") (collectively, the "License").  You
# may not use this file except in compliance with the License.  You can
# Oracle designates this particular file as subject to the "Classpath"
# exception as provided by Oracle in the GPL Version 2 section of the License
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#     http://www.apache.org/licenses/LICENSE-2.0
# distributed under the License is distributed on an "AS IS" BASIS,
```

Found in path(s):
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/PrivateMessages.properties
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
```

* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ValueReference.java
No license file was found, but licenses were detected in source scan.

```
 subject to such option by the copyright
* holder.
*
*
* This file incorporates work covered by the following copyright and
* permission notice:
*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

```
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/ArrayELResolver.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/MapELResolver.java
*
 /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/MethodExpression.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ELResolver.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/ELContextListener.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/FactoryFinder.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ELUtil.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/VariableMapper.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/MethodInfo.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/PropertyNotWritableException.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ListELResolver.java
*
 /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/CompositeELResolver.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ELException.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/ValueExpression.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-
jar/javax/el/ExpressionFactory.java
```

* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ELContextEvent.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ELContext.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/FunctionMapper.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/MethodNotFoundException.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/ResourceBundleELResolver.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/PropertyNotFoundException.java
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/Expression.java

No license file was found, but licenses were detected in source scan.

<!--

DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Copyright (c) 1997-2011 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can obtain a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html or packager/legal/LICENSE.txt. See the License for the specific language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields enclosed by brackets [] replaced by your own identifying information: "Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or only the GPL Version 2, indicate your decision by adding "[Contributor] elects to include this software in this distribution under the [CDDL or GPL Version 2] license." If you don't indicate a single choice of license, a

Found in path(s):
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/pom.xml
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175363270_1695326820.9279668/0/javax-el-api-2-2-4-sources-jar/javax/el/package.html

1.93 spring-expression 4.3.5

1.93.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ExpressionParser.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpEQ.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/ReflectivePropertyAccessor.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/SpelMessage.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/SpelParseException.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/BooleanLiteral.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpNE.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/Tokenizer.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/TokenKind.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/FloatLiteral.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/PropertyOrFieldReference.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/CodeFlow.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/BeanResolver.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OperatorInstanceof.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/ReflectionHelper.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/EvaluationException.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/InternalSpelExpressionParser.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/SpelEvaluationException.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/StandardTypeConverter.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/IntLiteral.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ParseException.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/LongLiteral.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ExpressionException.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/FunctionReference.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Indexer.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Operator.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2015 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/

Found in path(s):
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/ReflectiveMethodExecutor.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpPlus.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/ReflectiveMethodResolver.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpModulus.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/ConstructorReference.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpMinus.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/CompoundExpression.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpDivide.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/MethodReference.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ExpressionState.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpMultiply.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OperatorMatches.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Selection.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Projection.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/SpelCompiler.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/RealLiteral.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/StandardEvaluationContext.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpGE.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/CompilablePropertyAccessor.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/TypedValue.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/QualifiedIdentifier.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/ReflectiveConstructorResolver.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OperatorPower.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/InlineMap.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpOr.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpLT.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/common/ExpressionUtils.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Ternary.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/ValueRef.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpDec.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/ReflectiveConstructorExecutor.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Elvis.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Literal.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/StringLiteral.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-

jar/org/springframework/expression/spel/ast/OpAnd.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/InlineList.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/SpelNode.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/CompiledExpression.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/SpelNodeImpl.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/Token.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/SpelExpression.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/NullLiteral.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/standard/SpelExpressionParser.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/TypeReference.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/SpelCompilerMode.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Identifier.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/StandardTypeComparator.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/BeanReference.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/AstUtils.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OperatorNot.java
*
 /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpGT.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/FormatHelper.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/StandardTypeLocator.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/SpelParserConfiguration.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/TypeCode.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/VariableReference.java
*

/opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/PropertyAccessor.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/Assign.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpLE.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OpInc.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/ast/OperatorBetween.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/TypeLocator.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/Operation.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/TypeComparator.java
*
/opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/Expression.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ConstructorResolver.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/TypeConverter.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/common/LiteralExpression.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/common/TemplateParserContext.java
* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/OperatorOverloader.java

*

/opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/common/CompositeStringExpression.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/MethodResolver.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/common/TemplateAwareExpressionParser.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ParserContext.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/MethodExecutor.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/BooleanTypedValue.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/EvaluationContext.java

*

/opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ConstructorExecutor.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/ExpressionInvocationTargetException.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/MethodFilter.java

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/support/StandardOperatorOverloader.java

* /opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/spel/InternalParseException.java

*

/opt/cola/permits/175363977_1695324587.0296333/0/spring-expression-4-3-5-release-sources-jar/org/springframework/expression/AccessException.java

1.94 bson4jackson 2.7.0

1.94.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonRegexSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/deserializers/BsonDeserializers.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/ByteOrderUtil.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonJavaScriptSerializer.java
*
 /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/deserializers/BsonDeserializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonTimestampSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/types/ObjectId.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/DynamicOutputBuffer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/types/JavaScript.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/StaticBuffers.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/deserializers/BsonDateDeserializer.java
*
 /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/BoundedInputStream.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/types/Symbol.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonObjectIdSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/CountingInputStream.java

* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/LittleEndianInputStream.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/StaticBufferedInputStream.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/types/Timestamp.java
*
/opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonDateSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/io/UnsafeByteArrayInputStream.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/BsonModule.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonSymbolSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/deserializers/BsonCalendarDeserializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/BsonGenerator.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonUuidSerializer.java
*
/opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/BsonParser.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/BsonConstants.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/serializers/BsonCalendarSerializer.java
* /opt/cola/permits/175364775_1695324589.7932274/0/bson4jackson-2-7-0-sources-zip/de/undercouch/bson4jackson/BsonFactory.java

1.95 spring-framework 4.3.5

1.95.1 Available under license :

No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/ClassWriter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Handle.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/MethodWriter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/ByteVector.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/AnnotationVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Context.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Opcodes.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Frame.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/MethodVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/ClassReader.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/FieldWriter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Type.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/asm/Item.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Label.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/AnnotationWriter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/ClassVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/FieldVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Handler.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Attribute.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/Edge.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2013 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/TypeReference.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/TypePath.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/AbstractTypeHierarchyTraversingFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxEventHandler.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/ClassMetadataReadingVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/asm/SpringAsmInfo.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/PropertiesLoaderSupport.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/backoff/BackOffExecution.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToCollectionConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ObjectToArrayConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/lang/UsesSunHttpServer.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ObjectToStringConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/core/convert/converter/ConditionalConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/IdToEntityConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/StandardReflectionParameterNameDiscoverer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/VfsUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/backoff/BackOff.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/CompositeIterator.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/SerializationUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/ClassMetadata.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/PropertiesPropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ExceptionTypeFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/lang/UsesJava8.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/CollectionToCollectionConverter.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ZoneIdToTimeZoneConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/PropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/NestedIOException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ArrayToStringConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/RecursiveAnnotationArrayVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ArrayToObjectConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/AbstractClassTestingTypeFilter.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/SimpleMetadataReaderFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/ListenableFutureTask.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/AbstractStaxHandler.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ConversionServiceFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/BridgeMethodResolver.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/MutablePropertySources.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/MapToMapConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/AssignableTypeFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/lang/UsesJava7.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ArrayToArrayConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/JOptCommandLinePropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/CustomizableThreadCreator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxStreamHandler.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxStreamXMLReader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ArrayToCollectionConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/ResourceArrayPropertyEditor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/ResourceEditor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/PropertyPlaceholderHelper.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/Log4jConfigurer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ZonedDateTimeToCalendarConverter.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/MapPropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/StandardClassMetadata.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/AspectJTypeFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/LocalVariableTableParameterNameDiscoverer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/PropertiesToStringConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/cglib/core/SpringNamingPolicy.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ObjectToCollectionConverter.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/XmlValidationModeDetector.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/cglib/SpringCglibInfo.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxResult.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ConfigurableConversionService.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/SimpleCommandLineArgsParser.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ConvertingPropertyEditorAdapter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/CommandLineArgs.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/MissingRequiredPropertiesException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/PropertySources.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ParameterizedTypeReference.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/RegexPatternTypeFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/ContextResource.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/style/ValueStyler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/ResourcePatternResolver.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/TypeFilter.java
No license file was found, but licenses were detected in source scan.

/*

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/FileCopyUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/SystemPropertyUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/SimpleCommandLinePropertySource.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/InvalidMimeTypeException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/XMLEventStreamReader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/StandardEnvironment.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/EnvironmentCapable.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxSource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/AsyncListenableTaskExecutor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/GenericCollectionTypeResolver.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/AsyncTaskExecutor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/SimpleMetadataReader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/DomUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/MethodInvoker.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/XMLEventStreamWriter.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/StaxEventXMLReader.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2010 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/ConversionException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/VfsPatternUtils.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2008 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/NamedThreadLocal.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/NestedExceptionUtils.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/NamedInheritableThreadLocal.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2002-2015 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/SimpleNamespaceContext.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/GenericTypeResolver.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/support/SerializationDelegate.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/ResourcePropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/support/SerializingConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/DefaultDeserializer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ByteBufferConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/PatternMatchUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/SimpleIdGenerator.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/UpdateMessageDigestInputStream.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/converter/GenericConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/support/DeserializingConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/util/comparator/InstanceComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/InputStreamResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/filter/AnnotationTypeFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/Ordered.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/AnnotationMetadata.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/FallbackObjectToStringConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/LinkedMultiValueMap.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/converter/Converter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToCurrencyConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/AliasRegistry.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/StandardMethodMetadata.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/SynthesizedAnnotationInvocationHandler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/converter/ConvertingComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StreamConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/VfsResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/SimpleAliasRegistry.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/ReadOnlySystemAttributesMap.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/AlternativeJdkIdGenerator.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/DefaultParameterNameDiscoverer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/ClassPathResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/JdkVersion.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/converter/ConditionalGenericConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToCharsetConverter.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/SynthesizedAnnotation.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/AnnotatedTypeMetadata.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AliasFor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/SystemEnvironmentPropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToLocaleConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/StandardAnnotationMetadata.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ResolvableTypeProvider.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/CompositePropertySource.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AnnotationConfigurationException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/MethodMetadataReadingVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/IdGenerator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/objenesis/SpringObjenesis.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/JdkIdGenerator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToUUIDConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/DefaultSerializer.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ResizableByteArrayOutputStream.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/ByteArrayResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/AnnotationMetadataReadingVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/MethodMetadata.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/PriorityOrdered.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/EnumerablePropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/Order.java
*

/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToTimeZoneConverter.java

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AnnotationUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/GenericConversionService.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ObjectUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ResolvableType.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/AbstractResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/EnumToStringConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/IntegerToEnumConverterFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/ConfigurableEnvironment.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/AbstractEnvironment.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/FileSystemResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/SpringProperties.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/ListenableFutureCallbackRegistry.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/core/convert/support/DefaultConversionService.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/InstanceFilter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/NumberUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ConversionUtils.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/style/ToStringCreator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/AbstractRecursiveAnnotationVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/AbstractConditionalEnumConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/Property.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/TaskDecorator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/FutureAdapter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AbstractAliasAwareAnnotationAttributeExtractor.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/MethodIntrospector.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/AbstractStaxXMLReader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/Resource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AnnotationAttributeExtractor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/RecursiveAnnotationAttributesVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/AbstractXMLReader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/DecoratingClassLoader.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/lang/UsesSunMisc.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ResourceUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/SerializableTypeWrapper.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/MethodParameter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/CompletableToListenableFutureAdapter.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/SynthesizingMethodParameter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/PropertySourceFactory.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/CommandLinePropertySource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/DefaultResourceLoader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/StreamUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/FastByteArrayOutputStream.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/AntPathMatcher.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/SuccessCallback.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/DecoratingProxy.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/AnnotationReadingVisitorUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/ConverterNotFoundException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/StringUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/DefaultPropertySourceFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ReflectionUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/support/TaskExecutorAdapter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/ProtocolResolver.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AnnotationAwareOrderComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/ListenableFutureAdapter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/PropertiesLoaderUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/ResourceLoader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ObjectToObjectConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/MapAnnotationAttributeExtractor.java
*

/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/Conventions.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/ListenableFutureCallback.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/StringValueResolver.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ConfigurableObjectInputStream.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/backoff/FixedBackOff.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/OverridingClassLoader.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/AbstractPropertyResolver.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/EncodedResource.java

*

/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/converter/ConverterRegistry.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/TypeDescriptor.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/Assert.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/ObjectToOptionalConverter.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/CollectionUtils.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/UrlResource.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/ConversionService.java

*

/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/DefaultAnnotationAttributeExtractor.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/ResourceRegion.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/ListenableFuture.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/AutoPopulatingList.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AnnotationAttributes.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/Base64Utils.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/SocketUtils.java

*

/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/OrderUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToNumberConverterFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/ConversionFailedException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/DefaultPropertiesPersister.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/backoff/ExponentialBackOff.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/AbstractFileResolvingResource.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ConcurrentReferenceHashMap.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/MimeTypeUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/AnnotationAttributesReadingVisitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/MethodClassKey.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/annotation/AnnotatedElementUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/converter/ConverterFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/PathMatchingResourcePatternResolver.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/PropertiesPersister.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/PropertyResolver.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/FailureCallback.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/CollectionToStringConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/ConfigurablePropertyResolver.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToEnumConverterFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/LinkedCaseInsensitiveMap.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/ResourcePatternUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/SimpleAsyncTaskExecutor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/util/StopWatch.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/EnumToIntegerConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/concurrent/SettableListenableFuture.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/CollectionFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/OrderComparator.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/MimeType.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/PathResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ClassUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/TypeUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/Environment.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/NumberToNumberConverterFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/SpringFactoriesLoader.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/DigestUtils.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/style/DefaultValueStyler.java

* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/NumberToCharacterConverter.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/MultiValueMap.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/TaskTimeoutException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/support/ExecutorServiceAdapter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/NestedRuntimeException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ControlFlow.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToArrayConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/InfrastructureProxy.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/SmartClassLoader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/DomContentHandler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/support/LocalizedResourceHelper.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ErrorHandler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/SimpleSaxErrorHandler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/CharacterToNumberFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ControlFlowFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/TransformerUtils.java
*
 /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/CommonsLogWriter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/WeakReferenceMonitor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ErrorCoded.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/Constants.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/style/StylerUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/Deserializer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/core/type/classreading/CachingMetadataReaderFactory.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToCharacterConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/DescriptiveResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/comparator/ComparableComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/support/SerializationFailedException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/AttributeAccessorSupport.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/CollectionToObjectConverter.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/ConcurrencyThrottleSupport.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/comparator/BooleanComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ExceptionDepthComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/AttributeAccessor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/CollectionToArrayConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/WritableResource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/SimpleTransformErrorListener.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/PrioritizedParameterNameDiscoverer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/FileSystemResourceLoader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/style/DefaultToStringStyler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/xml/AbstractXMLStreamReader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/FileSystemUtils.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/comparator/CompoundComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/TaskExecutor.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/TaskRejectedException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-

jar/org/springframework/core/io/ClassRelativeResourceLoader.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/serializer/Serializer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/comparator/InvertibleComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/style/ToStringStyler.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ConstantException.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/NestedCheckedException.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToPropertiesConverter.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/MetadataReaderFactory.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/comparator/NullSafeComparator.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/SyncTaskExecutor.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/io/InputStreamSource.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/util/PathMatcher.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/convert/support/StringToBooleanConverter.java
*
/opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/ParameterNameDiscoverer.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/SpringVersion.java
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/task/support/ConcurrentExecutorAdapter.java
No license file was found, but licenses were detected in source scan.

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/type/classreading/MetadataReader.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002-2016 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175366671_1695324595.03161/0/spring-core-4-3-5-release-sources-jar/org/springframework/core/env/PropertySourcesPropertyResolver.java

1.96 okio 1.13.0

1.96.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/HashingSink.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Options.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/HashingSource.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Pipe.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Utf8.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/ForwardingTimeout.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/SegmentedByteString.java

No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one or more
*  contributor license agreements.  See the NOTICE file distributed with
*  this work for additional information regarding copyright ownership.
*  The ASF licenses this file to You under the Apache License, Version 2.0
*  (the "License"); you may not use this file except in compliance with
*  the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing, software
*  distributed under the License is distributed on an "AS IS" BASIS,
*  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*  See the License for the specific language governing permissions and
*  limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Base64.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/ByteString.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
```

* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Segment.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/DeflaterSink.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/RealBufferedSource.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/GzipSink.java
*
 /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Okio.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/AsyncTimeout.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Util.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/SegmentPool.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Timeout.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/BufferedSink.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/RealBufferedSink.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/GzipSource.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Buffer.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/BufferedSource.java
*
 /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Sink.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/ForwardingSink.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/InflaterSource.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/Source.java
* /opt/cola/permits/175376068_1695324622.8381565/0/okio-1-13-0-sources-jar/okio/ForwardingSource.java

1.97 cxf 2.7.3

1.97.1 Available under license :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="generator" content="HTML Tidy, see www.w3.org" />
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" type="text/css"
href="/StyleSheets/base.css" />
<title>W3C IPR SOFTWARE NOTICE</title>
```

```
</head>
<body text="#000000" bgcolor="#FFFFFF">
<h1>W3C<sup>&reg;</sup> SOFTWARE NOTICE AND LICENSE</h1>

<h3>Copyright &copy; 1994-2002 <a href="http://www.w3.org/">World
Wide Web Consortium</a>, (<a
href="http://www.lcs.mit.edu/">Massachusetts Institute of
Technology</a>, <a href="http://www.inria.fr/">Institut National de
Recherche en Informatique et en Automatique</a>, <a
href="http://www.keio.ac.jp/">Keio University</a>). All Rights
Reserved. http://www.w3.org/Consortium/Legal/</h3>

<p>This W3C work (including software, documents, or other related
items) is being provided
 by the copyright holders under the
following license. By obtaining, using and/or copying this work,
you (the licensee) agree that you have read, understood, and will
comply with the following terms and conditions:</p>

<p>Permission to use, copy, modify, and distribute this software
and its documentation, with or without modification,&nbsp; for any
purpose and without fee or royalty is hereby granted, provided that
you include the following on ALL copies of the software and
documentation or portions thereof, including modifications, that
you make:</p>

<ol>
<li>The full text of this NOTICE in a location viewable to users of
the redistributed or derivative work.</li>

<li>Any pre-existing intellectual property disclaimers, notices, or
terms and conditions. If none exist, a short notice of the
following form (hypertext is preferred, text is permitted) should
be used within the body of any redistributed or derivative code:
"Copyright &copy; [$date-of-software] <a
href="http://www.w3.org/">World
 Wide Web Consortium</a>, (<a
href="http://www.lcs.mit.edu/">Massachusetts Institute of
Technology</a>, <a href="http://www.inria.fr/">Institut National de
Recherche en Informatique et en Automatique</a>, <a
href="http://www.keio.ac.jp/">Keio University</a>). All Rights
Reserved. http://www.w3.org/Consortium/Legal/"</li>

<li>Notice of any changes or modifications to the W3C files,
including the date changes were made. (We recommend you provide
URIs to the location from which the code is derived.)</li>
</ol>
```

<p>THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND
COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE
USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD
PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.</p>

<p>COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT,
SPECIAL OR CONSEQUENTIAL DAMAGES
ARISING OUT OF ANY USE OF THE
SOFTWARE OR DOCUMENTATION.</p>

<p>The name and trademarks of copyright holders may NOT be used in
advertising or publicity pertaining to the software without
specific, written prior permission. Title to copyright in this
software and any associated documentation will at all times remain
with copyright holders.</p>

<p>___________________________________</p>

<p>This formulation of W3C's notice and license became active on
August 14 1998 so as to improve compatibility with GPL. This
version ensures that W3C software licensing terms are no more
restrictive than GPL and consequently W3C software may be
distributed in GPL packages. See the <a
href="copyright-software-19980519.html">older formulation</a> for
the policy prior to this date. Please see our <a
href="IPR-FAQ.html">Copyright FAQ</a> for common questions about
using materials from our site, including specific terms and
conditions for packages like libwww, Amaya, and Jigsaw. Other
questions about this
notice can be directed to <a
href="mailto:site-policy@w3.org">site-policy@w3.org</a>.<br />
&nbsp;</p>

<p>&nbsp;</p>

<address><a href="../../Help/Webmaster.html">webmaster</a><br />
(last updated $Date: 2006-11-08 08:52:14 -0500 (Wed, 08 Nov 2006) $)</address>
</body>
</html>
Copyright (c) 2004-2007 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2000-2005 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO
EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required
to implement this specification. Please address the information to the OASIS Executive Director.

Copyright OASIS Open 2002-2004. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself does not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked
by OASIS or its successors or assigns.

This document and the information contained herein is provided on an AS IS basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Eclipse Public License - Version 1.0</title>
<style type="text/css">
 body {
   size: 8.5in 11.0in;
   margin: 0.25in 0.5in 0.25in 0.5in;
   tab-interval: 0.5in;
   }
 p {
   margin-left: auto;
   margin-top:  0.5em;
   margin-bottom: 0.5em;
   }
 p.list {
  margin-left: 0.5in;
   margin-top:  0.05em;
   margin-bottom: 0.05em;
   }
 </style>
```

```
</head>

<body lang="EN-US">

<h2>Eclipse Public License - v 1.0</h2>

<p>THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE
PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR
DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS
AGREEMENT.</p>

<p><b>1. DEFINITIONS</b></p>

<p>"Contribution" means:</p>

<p
 class="list">a) in the case of the initial Contributor, the initial
code and documentation distributed under this Agreement, and</p>
<p class="list">b) in the case of each subsequent Contributor:</p>
<p class="list">i) changes to the Program, and</p>
<p class="list">ii) additions to the Program;</p>
<p class="list">where such changes and/or additions to the Program
originate from and are distributed by that particular Contributor. A
Contribution 'originates' from a Contributor if it was added to the
Program by such Contributor itself or anyone acting on such
Contributor's behalf. Contributions do not include additions to the
Program which: (i) are separate modules of software distributed in
conjunction with the Program under their own license agreement, and (ii)
are not derivative works of the Program.</p>

<p>"Contributor" means any person or entity that distributes
the Program.</p>

<p>"Licensed Patents" mean patent claims licensable by a
Contributor which are necessarily
 infringed by the use or sale of its
Contribution alone or when combined with the Program.</p>

<p>"Program" means the Contributions distributed in accordance
with this Agreement.</p>

<p>"Recipient" means anyone who receives the Program under
this Agreement, including all Contributors.</p>

<p><b>2. GRANT OF RIGHTS</b></p>

<p class="list">a) Subject to the terms of this Agreement, each
Contributor hereby grants Recipient a non-exclusive, worldwide,
```

royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.</p>

<p class="list">b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.</p>

<p class="list">c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.</p>

<p class="list">d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.</p>

<p><b>3. REQUIREMENTS</b></p>

<p>A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:</p>

<p class="list">a) it complies with the terms and conditions of this Agreement; and</p>

<p class="list">b) its license agreement:</p>

<p class="list">i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties

or conditions of title and non-infringement, and implied warranties or
conditions of merchantability and fitness for
 a particular purpose;</p>

<p class="list">ii) effectively excludes on behalf of all Contributors
all liability for damages, including direct, indirect, special,
incidental and consequential damages, such as lost profits;</p>

<p class="list">iii) states that any provisions which differ from this
Agreement are offered by that Contributor alone and not by any other
party; and</p>

<p class="list">iv) states that source code for the Program is available
from such Contributor, and informs licensees how to obtain it in a
reasonable manner on or through a medium customarily used for software
exchange.</p>

<p>When the Program is made available in source code form:</p>

<p class="list">a) it must be made available under this Agreement; and</p>

<p class="list">b) a copy of this Agreement must be included with each
copy of the Program.</p>

<p>Contributors may not remove or alter any copyright notices contained
within the Program.</p>

<p>Each Contributor must identify itself as the originator
 of its
Contribution, if any, in a manner that reasonably allows subsequent
Recipients to identify the originator of the Contribution.</p>

<p><b>4. COMMERCIAL DISTRIBUTION</b></p>

<p>Commercial distributors of software may accept certain
responsibilities with respect to end users, business partners and the
like. While this license is intended to facilitate the commercial use of
the Program, the Contributor who includes the Program in a commercial
product offering should do so in a manner which does not create
potential liability for other Contributors. Therefore, if a Contributor
includes the Program in a commercial product offering, such Contributor
("Commercial Contributor") hereby agrees to defend and
indemnify every other Contributor ("Indemnified Contributor")
against any losses, damages and costs (collectively "Losses")
arising from claims, lawsuits and other legal actions brought by a third
party against the Indemnified Contributor to the extent caused by the
acts or omissions

of such Commercial Contributor in connection with its
distribution of the Program in a commercial product offering. The
obligations in this section do not apply to any claims or Losses
relating to any actual or alleged intellectual property infringement. In
order to qualify, an Indemnified Contributor must: a) promptly notify
the Commercial Contributor in writing of such claim, and b) allow the
Commercial Contributor to control, and cooperate with the Commercial
Contributor in, the defense and any related settlement negotiations. The
Indemnified Contributor may participate in any such claim at its own
expense.</p>

<p>For example, a Contributor might include the Program in a commercial
product offering, Product X. That Contributor is then a Commercial
Contributor. If that Commercial Contributor then makes performance
claims, or offers warranties related to Product X, those performance
claims and warranties are such Commercial Contributor's responsibility
alone. Under this section, the
Commercial Contributor would have to
defend claims against the other Contributors related to those
performance claims and warranties, and if a court requires any other
Contributor to pay any damages as a result, the Commercial Contributor
must pay those damages.</p>

<p><b>5. NO WARRANTY</b></p>

<p>EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS
PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION,
ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely
responsible for determining the appropriateness of using and
distributing the Program and assumes all risks associated with its
exercise of rights under this Agreement , including but not limited to
the risks and costs of program errors, compliance with applicable laws,
damage to or loss of data, programs or equipment, and unavailability or
interruption of operations.</p>

<p><b>6.
DISCLAIMER OF LIABILITY</b></p>

<p>EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT
NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING
WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.</p>

<p><b>7. GENERAL</b></p>

<p>If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.</p>

<p>If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.</p>

<p>All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.</p>

<p>Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.</p>

<p>This Agreement is governed by the laws of the State of New York and
the intellectual property laws of the United States of America. No party
to this Agreement will bring a legal action under this Agreement more
than one year after the cause of action arose. Each party waives its
rights to a jury trial in any resulting litigation.</p>

</body>

</html>

Portions of the file cxf-utils.js derives from code marked:
This code was written by Tyler Akins and has been placed in the
public domain. It would be nice if you left this header intact.
Base64 code from Tyler Akins -- http://rumkin.com

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only

Apache CXF includes a number of components and libraries with separate copyright notices and license terms. Your use of those components are subject to the terms and conditions of the following licenses.

AntLR Parser Generator (http://www.antlr.org/) antlr:antlr:jar:2.7.7
License: BSD License (http://www.antlr.org/license.html)

AOP alliance (http://aopalliance.sourceforge.net) aopalliance:aopalliance:jar:1.0
License: Public Domain

ASM (http://asm.objectweb.org/asm/asm) asm:asm:jar:3.3.1:compile
License: BSD (http://asm.ow2.org/license.html)

MSV XML Schema Datatype Library (http://nexus.sonatype.org/oss-repository-hosting.html/xsdlib)
com.sun.msv.datatype.xsd:xsdlib:bundle:2010.1
License: BSD (http://www.opensource.org/licenses/bsd-license.php)

Sun JAXB Reference Implementation Runtime (http://jaxb.java.net/jaxb-impl) com.sun.xml.bind:jaxb-impl:jar:2.2.6:compile
License: COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0 (http://www.sun.com/cddl/cddl.html)

Sun JAXB Reference Implementation Tools (http://jaxb.java.net/jaxb-xjc) com.sun.xml.bind:jaxb-xjc:jar:2.2.6:compile
License: COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0 (http://www.sun.com/cddl/cddl.html)

ISO Relax (http://iso-relax.sourceforge.net/) isorelax:isorelax:jar:20030108:compile
License: MIT (http://www.opensource.org/licenses/mit-license.html)

javax.ws.rs-api (http://jax-rs-spec.java.net) javax.ws.rs:javax.ws.rs-api:jar:2.0-m10
License: CDDL 1.1 (http://glassfish.java.net/public/CDDL+GPL_1_1.html)

javax.ws.rs-api (http://jax-rs-spec.java.net) javax.ws.rs:javax.ws.rs-api:jar:2.0-m10
License: GPL2 w/ CPE (http://glassfish.java.net/public/CDDL+GPL_1_1.html)

Java Architecture for XML Binding (JAXB API) (http://jaxb.java.net/jaxb-api) javax.xml.bind:jaxb-api:jar:2.2.6:compile
License: COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0 (http://www.sun.com/cddl/cddl.html)

Joda time (http://joda-time.sourceforge.net) joda-time:joda-time:jar:1.6.2
License: Apache 2 (http://www.apache.org/licenses/LICENSE-2.0.txt)

MSV Core (http://msv.java.net/msv-core) net.java.dev.msv:msv-core:bundle:2011.1
License: BSD ()

HttpAsyncClient (http://hc.apache.org/httpcomponents-asyncclient)
org.apache.httpcomponents:httpasyncclient:jar:4.0-beta3
License: Apache License (LICENSE.txt)

HttpClient (http://hc.apache.org/httpcomponents-client) org.apache.httpcomponents:httpclient:jar:4.2.1
License: Apache License (LICENSE.txt)

HttpCore (http://hc.apache.org/httpcomponents-core-ga) org.apache.httpcomponents:httpcore:jar:4.2.2
License:
Apache License (LICENSE.txt)

HttpCore NIO (http://hc.apache.org/httpcomponents-core-ga) org.apache.httpcomponents:httpcore-nio:jar:4.2.2
License: Apache License (LICENSE.txt)

Apache MINA Core (http://mina.apache.org/mina-core/) org.apache.mina:mina-core:bundle:2.0.5
License: Apache 2.0 License (http://www.apache.org/licenses/LICENSE-2.0)

Stax2 API (http://woodstox.codehaus.org/StAX2) org.codehaus.woodstox:stax2-api:jar:3.1.1
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

Jetty :: Continuation (http://www.eclipse.org/jetty/jetty-continuation) org.eclipse.jetty:jetty-continuation:jar:8.1.7.v20120910
License: Apache Software License - Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Jetty :: Continuation (http://www.eclipse.org/jetty/jetty-continuation) org.eclipse.jetty:jetty-continuation:jar:8.1.7.v20120910
License: Eclipse Public License - Version 1.0 (http://www.eclipse.org/org/documents/epl-v10.php)

Jetty :: Http
Utility (http://www.eclipse.org/jetty/jetty-http) org.eclipse.jetty:jetty-http:jar:8.1.7.v20120910
License: Apache Software License - Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Jetty :: Http Utility (http://www.eclipse.org/jetty/jetty-http) org.eclipse.jetty:jetty-http:jar:8.1.7.v20120910
License: Eclipse Public License - Version 1.0 (http://www.eclipse.org/org/documents/epl-v10.php)

Jetty :: IO Utility (http://www.eclipse.org/jetty/jetty-io) org.eclipse.jetty:jetty-io:jar:8.1.7.v20120910
License: Apache Software License - Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Jetty :: IO Utility (http://www.eclipse.org/jetty/jetty-io) org.eclipse.jetty:jetty-io:jar:8.1.7.v20120910
License: Eclipse Public License - Version 1.0 (http://www.eclipse.org/org/documents/epl-v10.php)

Jetty :: Security (http://www.eclipse.org/jetty/jetty-security) org.eclipse.jetty:jetty-security:jar:8.1.7.v20120910
License: Apache Software License - Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Jetty
:: Security (http://www.eclipse.org/jetty/jetty-security) org.eclipse.jetty:jetty-security:jar:8.1.7.v20120910
License: Eclipse Public License - Version 1.0 (http://www.eclipse.org/org/documents/epl-v10.php)

Jetty :: Server Core (http://www.eclipse.org/jetty/jetty-server) org.eclipse.jetty:jetty-server:jar:8.1.7.v20120910
License: Apache Software License - Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Jetty :: Server Core (http://www.eclipse.org/jetty/jetty-server) org.eclipse.jetty:jetty-server:jar:8.1.7.v20120910
License: Eclipse Public License - Version 1.0 (http://www.eclipse.org/org/documents/epl-v10.php)

Jetty :: Utilities (http://www.eclipse.org/jetty/jetty-util) org.eclipse.jetty:jetty-util:jar:8.1.7.v20120910
License: Apache Software License - Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Jetty :: Utilities (http://www.eclipse.org/jetty/jetty-util) org.eclipse.jetty:jetty-util:jar:8.1.7.v20120910
License: Eclipse Public License - Version 1.0 (http://www.eclipse.org/org/documents/epl-v10.php)

OpenSAML-J (http://opensaml.org/) org.opensaml:opensaml:jar:2.5.1-1
License: Apache 2 (http://www.apache.org/licenses/LICENSE-2.0.txt)

OpenWS (http://opensaml.org/) org.opensaml:openws:jar:1.4.2-1
License: Apache 2 (http://www.apache.org/licenses/LICENSE-2.0.txt)

XMLTooling-J (http://opensaml.org/) org.opensaml:xmltooling:jar:1.3.2-1
License: Apache 2 (http://www.apache.org/licenses/LICENSE-2.0.txt)

Simple Logging Facade for Java - API (http://www.slf4j.org/slf4j-api) org.slf4j:slf4j-api:jar:1.6.2:compile
License: MIT License (http://www.slf4j.org/license.html)

Simple Logging Facade for Java - JDK Logging (http://www.slf4j.org/slf4j-jdk14) org.slf4j:slf4j-jdk14:jar:1.6.2:compile
License: MIT License (http://www.slf4j.org/license.html)

RELAX NG Datatype (http://sourceforge.net/projects/relaxng/)
relaxngDatatype:relaxngDatatype:jar:20020414:compile
License: BSD (http://www.opensource.org/licenses/bsd-license.php)

Rhino (http://www.mozilla.org/rhino/) rhino:js:jar:1.7R2
License: Mozilla Public License (http://www.mozilla.org/MPL/MPL-1.1.html)

WSDL4J (http://sf.net/projects/wsdl4j) wsdl4j:wsdl4j:jar:1.6.2
License: CPL (http://www.opensource.org/licenses/cpl1.0.txt)

Permission to copy, display, perform, modify and distribute the WS-Addressing Specification, and to authorize others to do the foregoing, in any medium without fee or royalty is hereby granted for the purpose of developing and evaluating the WS-Addressing Specification.

BEA, IBM, Microsoft, SAP AG, and Sun Microsystems (collectively, the "Authors") each agree to grant a license to third parties, under royalty-free and otherwise reasonable, non-discriminatory terms and conditions, to their respective essential patent claims that they deem necessary to implement the WS-Addressing Specification.

DISCLAIMERS:

THE WS-Addressing Specification IS PROVIDED "AS IS", AND THE AUTHORS MAKE NO REPRESENTATIONS
OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE
CONTENTS OF THE WS-Addressing Specification IS SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS,
TRADEMARKS OR OTHER RIGHTS.

THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR

CONSEQUENTIAL
DAMAGES ARISING OUT OF ANY USE OF THE WS-Addressing Specification OR THE PERFORMANCE OR

IMPLEMENTATION OF THE CONTENTS THEREOF.

You may remove these disclaimers from your modified versions of the WS-Addressing Specification provided that you effectively disclaim all warranties and liabilities on behalf of all copyright holders in the copies of any such modified versions you distribute.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to the WS-Addressing Specification or its contents without specific, written prior permission. Title to copyright in the WS-Addressing Specification will at all times remain with the Authors.

No other rights are granted by implication, estoppel or otherwise.
Copyright 2001 - 2005, International Business Machines Corporation and Microsoft Corporation
All Rights Reserved

License for WSDL Schema Files

The Authors grant permission to copy and distribute the WSDL Schema Files in any medium without fee or royalty as long as this notice and license are distributed with them. The originals of these files can be located at:

http://schemas.xmlsoap.org/wsdl/2003-02-11.xsd

THESE SCHEMA FILES ARE PROVIDED "AS IS," AND THE AUTHORS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, REGARDING THESE FILES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OR TITLE. THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THESE FILES.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to these files or any program

or service that uses these files, written prior permission. Title to copyright in these files will at all times remain with the Authors.

No other rights are granted by implication, estoppel or otherwise.
This product also includes schemas and specification developed by:
- the W3C consortium (http://www.w3c.org)
(http://www.w3.org/XML/1998/namespace)

This product also includes WS-* schemas developed by International Business Machines Corporation, Microsoft Corporation, BEA Systems,

TIBCO Software, SAP AG, Sonic Software, and VeriSign
(http://schemas.xmlsoap.org/wsdl/2003-02-11.xsd)
(http://schemas.xmlsoap.org/ws/2004/08/addressing/)
(http://schemas.xmlsoap.org/wsdl/http)
(http://schemas.xmlsoap.org/ws/2005/02/rm/wsrm.xsd)
(http://www.w3.org/2005/08/addressing/ws-addr.xsd)
(http://www.w3.org/TR/ws-metadata-exchange/)
(http://schemas.xmlsoap.org/ws/2004/09/mex/)
(http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html)

The product contains code (StaxBuilder.java) that is
Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.
See the NOTICE.jdom file for additional information

Java classes (source and binary)
under org.apache.cxf.jaxws.javaee
are generated from schema available here:
(http://java.sun.com/xml/ns/javaee/javaee_5.xsd)

This Product also includes software developed by David Heinemeier Hansson.
(http://dev.rubyonrails.org/browser/trunk/activesupport/lib/active_support/inflections.rb)

This product includes software Copyright University of Southampton IT Innovation Centre, 2009
(http://www.it-innovation.soton.ac.uk).

This product also includes MTOSI wsdl and schemas developed by the
TeleManagement Forum (http://www.tmforum.org/browse.aspx). The original
MTOSI wsdl and schemas can be download from
(http://sourceforge.net/projects/mtosi-ri)

Portions of the included XmlSchema library are Copyright 2006 International Business Machines Corp.

Portions of the included xml-apis library were originally based on the following:
- software copyright (c) 1999, IBM Corporation., http://www.ibm.com.
- software copyright (c) 1999, Sun Microsystems., http://www.sun.com.
- software
copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

Portions of the included xmlbeans library were originally based on the following:
- software copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

Portions of the file cxf-utils.js derives from code marked:
This code was written by Tyler Akins and has been placed in the
public domain. It would be nice if you left this header intact.
Base64 code from Tyler Akins -- http://rumkin.com

Additional copyright notices and license terms applicable are present in the licenses directory of this distribution.
Copyright (c) 2005-2006 David Heinemeier Hansson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache CXF
Copyright 2006-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product also includes schemas and specification developed by:
- the W3C consortium (http://www.w3c.org)
(http://www.w3.org/XML/1998/namespace)

This product also includes WS-* schemas developed by International
Business Machines Corporation, Microsoft Corporation, BEA Systems,
TIBCO Software, SAP AG, Sonic Software, and VeriSign
(http://schemas.xmlsoap.org/wsdl/2003-02-11.xsd)
(http://schemas.xmlsoap.org/ws/2004/08/addressing/)
(http://schemas.xmlsoap.org/wsdl/http)
(http://schemas.xmlsoap.org/ws/2005/02/rm/wsrm.xsd)
(http://www.w3.org/2005/08/addressing/ws-addr.xsd)
(http://www.w3.org/TR/ws-metadata-exchange/)
(http://schemas.xmlsoap.org/ws/2004/09/mex/)
(http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html)

The product contains code (StaxBuilder.java) that is

See the NOTICE.jdom file for additional information

Java classes (source and binary) under org.apache.cxf.jaxws.javaee
are generated from schema available here:
(http://java.sun.com/xml/ns/javaee/javaee_5.xsd)

This Product also includes software developed by David Heinemeier Hansson.
(http://dev.rubyonrails.org/browser/trunk/activesupport/lib/active_support/inflections.rb)

This product includes software Copyright University of Southampton IT Innovation Centre, 2009
(http://www.it-innovation.soton.ac.uk).

This product also includes MTOSI wsdl and schemas developed by the
TeleManagement Forum (http://www.tmforum.org/browse.aspx). The original
MTOSI wsdl and schemas can be download from
(http://sourceforge.net/projects/mtosi-ri)

Portions of the included XmlSchema library are Copyright 2006 International Business Machines Corp.

Portions of the included xml-apis library were originally based on the
following:

- software copyright (c) 1999, IBM Corporation., http://www.ibm.com.
- software copyright (c) 1999, Sun Microsystems., http://www.sun.com.
- software copyright (c) 2000 World Wide Web Consortium, http://www.w3.org

Portions of the included xmlbeans library were originally based on the following:

- software copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

Portions of the file cxf-utils.js derives from code marked:
This code was written by Tyler Akins and has been placed in the
public domain. It would be nice if you left this header intact.
Base64 code from Tyler Akins -- http://rumkin.com

Additional copyright notices and license terms applicable are
present in the licenses directory of this distribution.
The BSD License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
##
## Licensed to the Apache Software Foundation (ASF) under one
## or more contributor license agreements.  See the NOTICE file
## distributed with this work for additional information
## regarding copyright ownership.  The ASF licenses this file
## to you under the Apache License, Version 2.0 (the
## "License"); you may not use this file except in compliance
## with the License.  You may obtain a copy of the License at
##
##  http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing,
## software distributed under the License is distributed on an
## "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
## KIND, either express or implied.  See the License for the
## specific language governing permissions and limitations
## under the License.
##
## $Date: 2008-03-09 23:17:06 -0700 (Sun, 09 Mar 2008) $ $Rev: 635446 $
##
```

Apache CXF includes a number of components and libraries with separate
copyright notices and license terms. Your use of
those components are
subject to the terms and conditions of the following licenses.

```
#set ( $apacheTxt = "The Apache Software License, Version 2.0" )

#foreach ( $project in $projects )
#foreach ( $license in $project.licenses)
#if ( ! ($apacheTxt == $license.name) )
```

$project.name #if ($project.url)($project.url)#end $project.artifact
   License: $license.name #if ($license.url) ($license.url)#end

#end
#end
#end

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition,
 we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.
Common Public License Version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative
works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents
 to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising
 the
rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability

and
fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent
Recipients to identify the
originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial
product
offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any

related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

```
/**
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/
package org.apache.cxf.rs.security.oauth2.common;

/**
* Base permission description
* @see OAuthAuthorizationData
*/
public class Permission {
   private String permission;
   private String description;
    private boolean isDefault;

   public Permission() {

   }

   public Permission(String permission, String description) {
       this.description = description;
       this.permission = permission;
   }

   /**
    * Gets the permission description
    * @return the description
    */
   public String getDescription() {
       return description;
   }

   /**
    * Sets the permission description
    * @param description
    */
   public void setDescription(String description) {
       this.description = description;
   }

   /**
```

```
     * Get the permission value such as "read_calendar"
     * @return the value
     */
    public String getPermission() {
        return permission;
    }

    /**
     * Sets the permission value such as "read_calendar"
     * @param permission the permission value
     */
    public void setPermission(String permission) {
        this.permission = permission;
    }

    /**
     * Indicates if this permission
 has been allocated by default or not.
     * Authorization View handlers may use this property in order to restrict
     * the list of scopes which may be refused to non-default scopes only.
     * For example, the read-only check-box controls can be used to represent
     * the default scopes
     * @param isDefault true if the permission has been allocated by default
     */
    public void setDefault(boolean value) {
        this.isDefault = value;
    }

    public boolean isDefault() {
        return isDefault;
    }
}
/*--
```

$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp $

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows

these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following:
"This product includes software developed by the
JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/
/**
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance

```
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

package org.apache.cxf.io;

import java.io.IOException;
import java.io.InputStream;

/**
* Marker interface for OutputStreams that can directly support
* copying from an input stream.  OutputStreams
 that maintain their
* own byte buffer or similar may be able to optimize the copy
* instead of using the read/write into a temporary buffer that
* the normal IOUtils.copy method requires.
*/
public interface CopyingOutputStream {

   int copyFrom(InputStream in) throws IOException;

}
```

The product contains code (StaxBuilder.java) that is
 Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
 All rights reserved.
 See the NOTICE.jdom file for additional information

```
/**
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/
package org.apache.cxf.rs.security.oauth.data;

/**
* Base permission description which is visible to
* authorization handlers
* @see OAuthAuthorizationData
*/
public class Permission {
   private
 String permission;
   private String description;
   private boolean isDefault;

   public Permission() {

   }

   public Permission(String permission, String description) {
       this.description = description;
       this.permission = permission;
   }

   public String getDescription() {
       return description;
   }

   public void setDescription(String description) {
       this.description = description;
   }

   public String getPermission() {
       return permission;
   }

   public void setPermission(String permission) {
       this.permission = permission;
   }

   /**
    * Indicates that this permission has been allocated by default.
    * Authorization View handlers may use this property in order to restrict
    * the list of scopes which may be refused to non-default scopes only
    * @param isDefault
```

```
     */
    public void setDefault(boolean value) {
        this.isDefault = value;
    }

    public boolean isDefault()
  {
        return isDefault;
    }
}
```

/*
$Id: LICENSE.txt,v 1.5 2006/02/05 21:49:04 elharo Exp $

Copyright 2003-2006 The Werken Company. All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

* Neither the name of the Jaxen Project nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

MOZILLA PUBLIC LICENSE
Version 1.1

---------------

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property

claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are
effective on the date Initial Developer first distributes
Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is
granted: 1) for code that You delete from the Original Code; 2)
separate from the Original Code; or 3) for infringements
caused
by: i) the modification of the Original Code or ii) the
combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor
hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or
trademark) Licensable by Contributor, to use, reproduce, modify,
display, perform, sublicense and distribute the Modifications
created by such Contributor (or portions thereof) either on an
unmodified basis, with other Modifications, as Covered Code
and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or
selling of Modifications made by that Contributor either alone
and/or in combination with its Contributor Version (or portions
of such combination), to make,
use, sell, offer for sale, have
made, and/or otherwise dispose of: 1) Modifications made by that
Contributor (or portions thereof); and 2) the combination of
Modifications made by that Contributor with its Contributor
Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are
effective on the date Contributor first makes Commercial Use of
the Covered Code.

(d)   Notwithstanding Section 2.2(b) above, no patent license is
granted: 1) for any code that Contributor has deleted from the
Contributor Version; 2) separate from the Contributor Version;
3) for infringements caused by: i) third party modifications of
Contributor Version or ii) the combination of Modifications made
by that Contributor with other software (except as part of the
Contributor Version) or other devices; or 4) under Patent Claims
infringed by Covered Code in the absence of Modifications made by
that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute
must be
made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement
that
the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
(a) Third Party Claims.
If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2,

Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by

this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or

collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename
Your license so that
the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer

or
a Contributor (the Initial Developer or Contributor against whom
You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly
infringes any patent, then any and all rights granted by such
Participant to You under Sections 2.1 and/or 2.2 of this License
shall, upon 60 days notice from Participant terminate prospectively,
unless if within 60 days after receipt of notice You either: (i)
agree in writing to pay Participant a mutually agreeable reasonable
royalty for Your past and future use of Modifications made by such
Participant, or (ii) withdraw Your litigation claim with respect to
the Contributor Version against such Participant. If within 60 days
of notice, a reasonable royalty and payment arrangement are not
mutually agreed upon in writing by the parties or the litigation claim
is not withdrawn, the rights granted by Participant to You under
Sections 2.1 and/or 2.2 automatically terminate at the expiration of
the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's
Contributor Version, directly or indirectly infringes any patent, then
any rights granted to You by such Participant under Sections 2.1(b)
and 2.2(b) are revoked effective as of the date You first made, used,
sold, distributed, or had made, Modifications made by that
Participant.

8.3. If You assert a patent infringement claim against Participant
alleging that such Participant's Contributor Version directly or
indirectly infringes any patent where such claim is resolved (such as
by license or settlement) prior to the initiation of patent
infringement litigation, then the reasonable value of the licenses
granted by such Participant under Sections 2.1 or 2.2 shall be taken
into account in determining the amount or value of any payment
or
license.

8.4. In the event of termination under Sections 8.1 or 8.2 above,
all end user license agreements (excluding distributors and resellers)
which have been validly granted by You or any distributor hereunder
prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL
DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE,
OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR
ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY

CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL
NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable,
such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be
construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly,

out of its utilization of rights under this License and You agree to
work with Initial Developer and Contributors to distribute such
responsibility on an equitable basis. Nothing herein is intended or
shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as
"Multiple-Licensed". "Multiple-Licensed" means that the Initial
Developer permits you to utilize portions of the Covered Code under
Your choice of the NPL or the alternative licenses, if any, specified
by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla
Public License
Version 1.1 (the "License"); you may not use this file except in
compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS"
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
License for the specific language governing rights and limitations
under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is ________________________.
Portions created by ______________________ are Copyright (C) ______
_______________________. All Rights Reserved.

Contributor(s): ______________________________________.

Alternatively, the contents of this file may be used under the terms
of the _____ license (the "[___] License"), in which case the
provisions of [______] License are applicable instead of those
above. If
you wish to allow use of your version of this file only
under the terms of the [____] License and not to allow others to use
your version of this file under the MPL, indicate your decision by
deleting the provisions above and replace them with the notice and
other provisions required by the [___] License. If you do not delete
the provisions above, a recipient may use your version of this file
under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of
the notices in the Source Code files of the Original Code. You should

use the text of this Exhibit A rather than the text found in the
Original Code Source Code for Your Modifications.]

Permission to copy and display the WS-Policy Specification (the "Specification", which includes WSDL and schema documents), in any medium without fee or royalty is hereby granted, provided that you include the following on ALL copies of the WS-Policy Specification, that you make:

1. A link or URL to the WS-Policy Specification at one of the Authors' websites
2. The copyright notice as shown in the WS-Policy Specification.

BEA Systems, IBM, Microsoft, SAP, Sonic Software, and VeriSign (collectively, the "Authors") each agree to grant you a license, under royalty-free and otherwise reasonable, non-discriminatory terms and conditions, to their respective essential patent claims that they deem necessary to implement the WS-Policy Specification.

THE WS-POLICY SPECIFICATION IS PROVIDED "AS IS," AND THE AUTHORS MAKE NO REPRESENTATIONS
OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE WS-POLICY SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE WS-POLICY SPECIFICATION.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to the WS-Policy Specification or its contents without specific, written prior permission. Title to copyright in the WS-Policy Specification will at all times remain with the Authors.

No other rights are granted by implication, estoppel or otherwise.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or

(b) Modifications, or (c) the combination of files
containing Original Software with files containing
Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form
other than Source Code.

1.5. Initial Developer means the individual or entity
that first makes Original Software available under this
License.

1.6. Larger Work means a work which combines Covered
Software or portions thereof with code not governed by the
terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the
maximum extent possible, whether at the time of the initial
grant or subsequently acquired, any and all of the rights
conveyed herein.

1.9. Modifications means the Source Code and Executable
form of any of the following:

A. Any file that results from an addition to,
deletion from or modification of the contents of a
file containing Original Software or previous
Modifications;

B. Any new file that contains any part of the
Original Software or previous Modification; or

C. Any new file that is contributed or otherwise
made
available under the terms of this License.

1.10. Original Software means the Source Code and
Executable form of computer software code that is
originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned
or hereafter acquired, including without limitation,
method, process, and apparatus claims, in any patent
Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with
You. For purposes of this
definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without
Modifications,
and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original

Software,
or (ii) the combination of the Original
Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications
made by that
Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2)
for
infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make
available. You must inform recipients of any
such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or
any descriptive text giving
attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by

You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer
or such Contributor as a result of
warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You

hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward
has the
right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3.
Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE
IN ANY RESPECT, YOU (NOT THE
INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO

NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause
is in lieu of, and supersedes,
any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without
limitation, court
costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or

indirectly, out of its utilization of rights under this License
and You agree to work with Initial Developer and Contributors to
distribute such responsibility on an equitable
basis. Nothing
herein is intended or shall be deemed to constitute any admission
of liability.
Java classes (source and binary) under org.apache.cxf.jaxws.javaee
are generated from schema available here:
(http://java.sun.com/xml/ns/javaee/javaee_5.xsd)

This product includes software Copyright University of Southampton IT Innovation Centre, 2009
(http://www.it-innovation.soton.ac.uk).
File json2.min.js has been placed in the public domain.

Portions of the file uuid.min.js derives from code marked:
/*
Math.uuid.js (v1.4)
http://www.broofa.com
mailto:robert@broofa.com

Copyright (c) 2010 Robert Kieffer
Dual licensed under the MIT and GPL licenses.
*/

1.98 cxf-rt 2.7.3

1.98.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.99 aspect-j 1.8.13

1.99.1 Available under license :

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works
of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents
to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to
identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product
offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance
claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL

HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

```
*
* Contributors:
*   Adrian Colyer   Initial implementation
* *******************************************************************/
package org.aspectj.internal.lang.annotation;
```

1.100 byte-buddy 1.7.10

1.100.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/***
* ASM XML Adapter
* Copyright (c) 2004-2011, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES,
 INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/SAXModuleAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/SAXAdapter.java

* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/Processor.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/SAXCodeAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/SAXAnnotationAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/ASMContentHandler.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/SAXFieldAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/SAXClassAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/asm-xml.dtd
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Constructs a new {@link ClassNode}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the {@link #ClassNode(int)}
   * version.
   *
   * @throws IllegalStateException
   *            If a subclass calls this constructor.
    */
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ClassNode.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
```

```
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Constructs a new {@link Textifier}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the {@link #Textifier(int)}
   * version.
   *
   * @throws IllegalStateException
   *             If a subclass calls this constructor.
    */
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/Textifier.java
No license file was found, but licenses were detected in source scan.

```
<!--
* ASM XML Adapter
* Copyright (c) 2004-2011, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/xml/package.html
No license file was found, but licenses were detected in source scan.

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/FieldNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/IntInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/TraceFieldVisitor.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/RemappingFieldAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ModuleOpenNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Context.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/LookupSwitchInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/TraceSignatureVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tools/ModuleInfoBndPlugin.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/TraceClassVisitor.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/AnnotationNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/Method.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Frame.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/SourceValue.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/LocalVariablesSorter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Type.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/InsnNode.java

* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/CheckModuleAdapter.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/CheckClassAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/RemappingMethodAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/AnnotationWriter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/ModuleTargetAttribute.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/CurrentFrame.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/StaticInitMerger.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/ClassWriter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/TypeAnnotationNode.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/ASMifier.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/signature/SignatureReader.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/ClassRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/AbstractInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/FieldVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/CheckFieldAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/SignatureRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Label.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tools/Retrofitter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/Subroutine.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/ClassReader.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/Value.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/AnalyzerException.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-

jar/net/bytebuddy/jar/asm/commons/TableSwitchGenerator.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/Interpreter.java
*
/opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/SimpleVerifier.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/SerialVersionUIDAdder.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Handler.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Edge.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/FieldWriter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/InnerClassNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/TryCatchBlockSorter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/MethodVisitor.java
*
/opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/TraceModuleVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/AnnotationRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/Remapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Handle.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Opcodes.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/ModuleRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/TraceMethodVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/Analyzer.java
*
/opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/BasicValue.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/FieldInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/InsnList.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/VarInsnNode.java

* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/InstructionAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/BasicVerifier.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/RemappingSignatureAdapter.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/signature/SignatureWriter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ModuleExportNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/IincInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ModuleNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/LabelNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/MethodInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/ModuleHashesAttribute.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/CheckMethodAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/RemappingAnnotationAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/AnnotationVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/CheckAnnotationAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/signature/SignatureVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/TableSwitchInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/ByteVector.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Attribute.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/ModuleVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/LdcInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/SourceInterpreter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/AdviceAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-

jar/net/bytebuddy/jar/asm/ClassVisitor.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/CodeSizeEvaluator.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/MethodRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/JumpInsnNode.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/LineNumberNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/SimpleRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/InvokeDynamicInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/FieldRemapper.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/RemappingClassAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/MethodWriter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/MethodNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/MultiANewArrayInsnNode.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/ModuleResolutionAttribute.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/LocalVariableNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ModuleRequireNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/BasicInterpreter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/FrameNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/TryCatchBlockNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/Item.java
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/ModuleWriter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/Printer.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ModuleProvideNode.java

* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/TypeInsnNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/SmallSet.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/ParameterNode.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/Frame.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/TypeReference.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/TypePath.java

No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Constructs a new {@link LocalVariableAnnotationNode}. <i>Subclasses must
   * not use this constructor</i>. Instead, they must use the
   * {@link #LocalVariableAnnotationNode(int, TypePath, LabelNode[], LabelNode[], int[], String)}
   * version.
   *


   * @param typeRef
   *            a reference to the annotated type. See {@link TypeReference}.
   * @param typePath
   *            the path to the annotated type argument, wildcard bound, array
   *            element type, or static inner type within 'typeRef'. May be
   *            <tt>null</tt> if the annotation targets 'typeRef' as a whole.
   * @param start
   *            the fist instructions corresponding to the continuous ranges
   *            that make the scope of this local variable (inclusive).
   * @param end
   *            the last instructions corresponding to the continuous ranges
   *           that make the scope of this local variable (exclusive). This
   *           array must have the same size as the 'start' array.
   * @param index
   *           the local variable's index in each range. This array must have
   *           the same size as the 'start' array.
   * @param desc
   *           the class descriptor of
 the annotation class.
   */
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/LocalVariableAnnotationNode.java
No license file was found, but licenses were detected in source scan.

```
/**
    * Creates a new JSRInliner. <i>Subclasses must not use this
    * constructor</i>. Instead, they must use the
```

```
  * {@link #JSRInlinerAdapter(int, MethodVisitor, int, String, String, String, String[])}
  * version.
  *
  * @param mv
  *
 the <code>MethodVisitor</code> to send the resulting inlined
  *          method code to (use <code>null</code> for none).
  * @param access
  *          the method's access flags (see {@link Opcodes}). This
  *          parameter also indicates if the method is synthetic and/or
  *          deprecated.
  * @param name
  *          the method's name.
  * @param desc
  *          the method's descriptor (see {@link Type}).
  * @param signature
  *          the method's signature. May be <tt>null</tt>.
  * @param exceptions
  *          the internal names of the method's exception classes (see
  *          {@link Type#getInternalName() getInternalName}). May be
  *          <tt>null</tt>.
  * @throws IllegalStateException
  *           If a subclass calls this constructor.
  */
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/JSRInlinerAdapter.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```

```
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Creates a new {@link AnalyzerAdapter}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the
   * {@link #AnalyzerAdapter(int, String, int, String, String, MethodVisitor)}
   * version.
   *
   * @param owner
   *
     the owner's class name.
   * @param access
   *          the method's access flags (see {@link Opcodes}).
   * @param name
   *          the method's name.
   * @param desc
   *          the method's descriptor (see {@link Type Type}).
   * @param mv
   *          the method visitor to which this adapter delegates calls. May
   *          be <tt>null</tt>.
   * @throws IllegalStateException
   *           If a subclass calls this constructor.
   */
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/AnalyzerAdapter.java
No license file was found, but licenses were detected in source scan.

```
<!--
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
```

*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
-->

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/package.html
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/package.html
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/tree/analysis/package.html
*
 /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/signature/package.html
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/package.html
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/package.html
No license file was found, but licenses were detected in source scan.

/**
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/ASMifiable.java
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/util/Textifiable.java
No license file was found, but licenses were detected in source scan.

```
/**
  * Creates a new {@link GeneratorAdapter}. <i>Subclasses must not use this
  * constructor</i>. Instead, they must use the
  * {@link #GeneratorAdapter(int, MethodVisitor, int, String, String)}
  * version.
  *
  * @param mv
  *          the
 method visitor to which this adapter delegates calls.
  * @param access
  *          the method's access flags (see {@link Opcodes}).
  * @param name
  *          the method's name.
  * @param desc
  *          the method's descriptor (see {@link Type Type}).
  * @throws IllegalStateException
  *           If a subclass calls this constructor.
  */
```

Found in path(s):
* /opt/cola/permits/175407863_1695324648.5248663/0/byte-buddy-1-7-10-sources-jar/net/bytebuddy/jar/asm/commons/GeneratorAdapter.java

1.101 javatuples 1.2

1.101.1 Available under license :

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions

1.102 javax-el 2.2.4

1.102.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

exception as provided by Oracle in the GPL Version 2 section of the License
file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields
enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license." If you don't indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above. However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if
the new code is made subject to such option by the copyright
holder.

-->

Found in path(s):
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2011 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception

```
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

/*
 Author: Jacob Hookom
 Email: jacob at hookom.net
      Author: Kin-man Chung
*/

/* == Option Declaration == */
options
{
 STATIC=false;
 NODE_PREFIX="Ast";
 VISITOR_EXCEPTION="javax.el.ELException";
 VISITOR=false;
 MULTI=true;
 NODE_DEFAULT_VOID=true;
 JAVA_UNICODE_ESCAPE=false;
  UNICODE_INPUT=true;
 BUILD_NODE_FILES=true;
}

/* == Parser Declaration == */
PARSER_BEGIN( ELParser )
package com.sun.el.parser;
import java.io.StringReader;
import javax.el.ELException;
public class ELParser
```

```
{
   public static Node parse(String ref) throws ELException
   {
       try {
        return (new ELParser(new StringReader(ref))).CompositeExpression();
       } catch (ParseException pe) {
           throw new ELException(pe.getMessage());
       }
   }
}
PARSER_END( ELParser )

/*
* CompositeExpression
* Allow most flexible parsing, restrict by examining
* type of returned node
*/
AstCompositeExpression CompositeExpression() #CompositeExpression
 : {}
{
 (DeferredExpression() | DynamicExpression() | LiteralExpression())* <EOF> { return jjtThis; }
}

/*
* LiteralExpression
* Non-EL Expression blocks
*/
void LiteralExpression() #LiteralExpression : { Token t = null; }
{
 t=<LITERAL_EXPRESSION> { jjtThis.setImage(t.image); }
}

/*
* DeferredExpression
* #{..} Expressions
*/
void DeferredExpression() #DeferredExpression : {}
{
 <START_DEFERRED_EXPRESSION> Expression() <END_EXPRESSION>
}

/*
* DynamicExpression
* ${..} Expressions
*/
void DynamicExpression() #DynamicExpression : {}
{
 <START_DYNAMIC_EXPRESSION> Expression() <END_EXPRESSION>
```

```
}

/*
* Expression
* EL Expression Language Root, goes to Choice
*/
void Expression() : {}
{
 Choice()
}

/*
* Choice
* For Choice markup a ? b : c, then Or
*/
void Choice() : {}
{
 Or() (<QUESTIONMARK> Choice() <COLON> Choice() #Choice(3))?
}

/*
* Or
* For 'or' '||', then And
*/
void Or() : {}
{
 And() ((<OR0>|<OR1>) And() #Or(2))*
}

/*
* And
* For 'and'
 '&&', then Equality
*/
void And() : {}
{
 Equality() ((<AND0>|<AND1>) Equality() #And(2))*
}

/*
* Equality
* For '==' 'eq' '!=' 'ne', then Compare
*/
void Equality() : {}
{
 Compare()
 (
  ((<EQ0>|<EQ1>) Compare() #Equal(2))
 |
  ((<NE0>|<NE1>) Compare() #NotEqual(2))
 )*
}

/*
* Compare
* For a bunch of them, then Math
*/
void Compare() : {}
{
 Math()
 (
  ((<LT0>|<LT1>) Math() #LessThan(2))
 |
  ((<GT0>|<GT1>) Math() #GreaterThan(2))
 |
  ((<LE0>|<LE1>) Math() #LessThanEqual(2))
 |
  ((<GE0>|<GE1>) Math() #GreaterThanEqual(2))
 )*
}

/*
* Math
* For '+' '-', then Multiplication
*/
void Math() : {}
{
 Multiplication()
 (
  (<PLUS> Multiplication() #Plus(2))
 |
  (<MINUS> Multiplication() #Minus(2))
 )*
}

/*
* Multiplication
* For a bunch of them, then Unary
*/
void Multiplication() : {}
{
 Unary()
 (
  (<MULT> Unary() #Mult(2))
 |
  ((<DIV0>|<DIV1>) Unary() #Div(2))
 |
  ((<MOD0>|<MOD1>) Unary() #Mod(2))
 )*
}

/*
* Unary
* For '-' '!' 'not' 'empty', then Value
*/
void
 Unary() : {}
{
  <MINUS> Unary() #Negative
 |
  (<NOT0>|<NOT1>) Unary() #Not
 |
  <EMPTY> Unary() #Empty
 |
  Value()
}

/*
* Value
* Defines Prefix plus zero or more Suffixes
*/
void Value() : {}
{
 (ValuePrefix() (ValueSuffix())*) #Value(>1)
}

/*
* ValuePrefix
* For Literals, Variables, and Functions
*/
void ValuePrefix() : {}
{
 Literal()
 | NonLiteral()
}

/*
* ValueSuffix
* Either dot or bracket notation
*/
void ValueSuffix() : {}
{
 DotSuffix() | BracketSuffix()
}
```

```
/*
* DotSuffix
* Dot Property and Dot Method
*/
void DotSuffix() #DotSuffix : { Token t = null; }
{
       <DOT> t=<IDENTIFIER> { jjtThis.setImage(t.image); }
       ( (<LPAREN> (Expression() (<COMMA> Expression())*)? <RPAREN> )
           #MethodArguments
       )?
}

/*
* BracketSuffix
* Sub Expression Suffix
*/
void BracketSuffix() #BracketSuffix : {}
{
 <LBRACK> Expression() <RBRACK>
       ( (<LPAREN> (Expression() (<COMMA> Expression())*)? <RPAREN> )
           #MethodArguments

     )?
}

/*
* NonLiteral
* For Grouped Operations, Identifiers, and Functions
*/
void NonLiteral() : {}
{
 <LPAREN> Expression() <RPAREN>
 | LOOKAHEAD(4) Function()
 | Identifier()
}

/*
* Identifier
* Java Language Identifier
*/
void Identifier() #Identifier : { Token t = null; }
{
 t=<IDENTIFIER> { jjtThis.setImage(t.image); }
}

/*
* Function
* Namespace:Name(a,b,c)
*/
void Function() #Function :
{
 Token t0 = null;
 Token t1 = null;
}
{
 t0=<IDENTIFIER> (<COLON> t1=<IDENTIFIER>)?
 {
  if (t1 != null) {
   jjtThis.setPrefix(t0.image);
   jjtThis.setLocalName(t1.image);
  } else {
   jjtThis.setLocalName(t0.image);
  }
 }
 <LPAREN> (Expression() (<COMMA> Expression())*)? <RPAREN>
}


/*
* Literal
* Reserved Keywords
*/
void Literal() : {}
{
 Boolean()
 | FloatingPoint()
 | Integer()
 | String()
 | Null()
}

/*
* Boolean
* For 'true' 'false'
*/
void Boolean() : {}
{
 <TRUE> #True
 | <FALSE> #False
}

/*
* FloatinPoint
* For Decimal and Floating Point Literals
*/
```

```
void
 FloatingPoint() #FloatingPoint : { Token t = null; }
{
 t=<FLOATING_POINT_LITERAL> { jjtThis.setImage(t.image); }
}

/*
* Integer
* For Simple Numeric Literals
*/
void Integer() #Integer : { Token t = null; }
{
 t=<INTEGER_LITERAL> { jjtThis.setImage(t.image); }
}

/*
* String
* For Quoted Literals
*/
void String() #String : { Token t = null; }
{
 t=<STRING_LITERAL> { jjtThis.setImage(t.image); }
}

/*
* Null
* For 'null'
*/
void Null() #Null : {}
{
 <NULL>
}


/*
===============================================================================
== */
<DEFAULT> TOKEN :
{
 < LITERAL_EXPRESSION:
   ((~["\\", "$", "#"])
     | ("\\" ("\\" | "$" | "#"))
     | ("$" ~["{", "$"])
     | ("#" ~["{", "#"])
   )+
   | "$"
   | "#"
 >
```

```
|
 < START_DYNAMIC_EXPRESSION: "${" > : IN_EXPRESSION
|
 < START_DEFERRED_EXPRESSION: "#{" > : IN_EXPRESSION
}

<DEFAULT> SKIP : { "\\" }

<IN_EXPRESSION> SKIP : { " " | "\t" | "\n" | "\r" }

<IN_EXPRESSION> TOKEN :
{
 < INTEGER_LITERAL: ["0"-"9"]
 (["0"-"9"])* >
| < FLOATING_POINT_LITERAL: (["0"-"9"])+ "." (["0"-"9"])* (<EXPONENT>)?
  | "." (["0"-"9"])+ (<EXPONENT>)?
  | (["0"-"9"])+ <EXPONENT>
 >
| < #EXPONENT: ["e","E"] (["+","-"])? (["0"-"9"])+ >
| < STRING_LITERAL: ("\"" ((~["\"","\\"])
  | ("\\" ( ["\\","\""] )))* "\"")
  | ("\'" ((~["\'","\\"])
  | ("\\" ( ["\\","\'"] )))* "\'")
 >
| < BADLY_ESCAPED_STRING_LITERAL: ("\"" (~["\"","\\"])* ("\\" ( ~["\\","\""] )))
  | ("\'" (~["\'","\\"])* ("\\" ( ~["\\","\'"] )))
 >
| < TRUE : "true" >
| < FALSE : "false" >
| < NULL : "null" >
| < END_EXPRESSION : "}" > : DEFAULT
| < DOT : "." >
| < LPAREN : "(" >
| < RPAREN : ")" >
| < LBRACK : "[" >
| < RBRACK : "]" >
| < COLON : ":" >
| < COMMA : "," >
| < GT0 : ">" >
| < GT1 : "gt" >
| < LT0 : "<" >
| < LT1 : "lt" >
| < GE0 : ">=" >
| < GE1 : "ge" >
| < LE0 : "<=" >
| < LE1 : "le" >
| < EQ0 : "==" >
| < EQ1 : "eq" >
| < NE0 : "!=" >
| < NE1 : "ne" >
| < NOT0 : "!" >
| < NOT1 : "not" >
| < AND0 : "&&" >
| < AND1 : "and" >
| < OR0 :
 "||" >
| < OR1 : "or" >
| < EMPTY : "empty" >
| < INSTANCEOF : "instanceof" >
| < MULT : "*" >
| < PLUS : "+" >
| < MINUS : "-" >
| < QUESTIONMARK : "?" >
| < DIV0 : "/" >
| < DIV1 : "div" >
| < MOD0 : "%" >
| < MOD1 : "mod" >
| < IDENTIFIER : (<LETTER>|<IMPL_OBJ_START>) (<LETTER>|<DIGIT>)* >
| < #IMPL_OBJ_START: "#" >
| < #LETTER:
  [
  "\u0024",
  "\u0041"-"\u005a",
  "\u005f",
  "\u0061"-"\u007a",
  "\u00c0"-"\u00d6",
  "\u00d8"-"\u00f6",
  "\u00f8"-"\u00ff",
  "\u0100"-"\u1fff",
  "\u3040"-"\u318f",
  "\u3300"-"\u337f",
  "\u3400"-"\u3d2d",
  "\u4e00"-"\u9fff",
  "\uf900"-"\ufaff"
  ]
 >
| < #DIGIT:
  [
  "\u0030"-"\u0039",
  "\u0660"-"\u0669",
  "\u06f0"-"\u06f9",
  "\u0966"-"\u096f",
  "\u09e6"-"\u09ef",
  "\u0a66"-"\u0a6f",
  "\u0ae6"-"\u0aef",
  "\u0b66"-"\u0b6f",
  "\u0be7"-"\u0bef",
  "\u0c66"-"\u0c6f",
  "\u0ce6"-"\u0cef",
  "\u0d66"-"\u0d6f",
  "\u0e50"-"\u0e59",
  "\u0ed0"-"\u0ed9",
  "\u1040"-"\u1049"
  ]
 >
| < ILLEGAL_CHARACTER: (~[]) >
}
```

Found in path(s):
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ELParser.jjt
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/Messages.properties
No license file was found, but licenses were detected in source scan.

as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/SimpleCharStream.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/ExpressionBuilder.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstValue.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ELParserConstants.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ELParserTokenManager.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ELParserTreeConstants.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ELParser.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/JJTELParserState.java
*
/opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstMethodArguments.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 1997-2010 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU

* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/FunctionMapperFactory.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstNotEqual.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstNull.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/Token.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/VariableMapperImpl.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstDynamicExpression.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-

jar/com/sun/el/parser/AstLessThan.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstLiteralExpression.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/ValueExpressionLiteral.java
*
 /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/BooleanNode.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstDiv.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstDeferredExpression.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstEqual.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstFunction.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstMinus.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstPlus.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/MethodExpressionLiteral.java
*
 /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstNot.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstIdentifier.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/SimpleNode.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/MethodExpressionImpl.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/VariableMapperFactory.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstFalse.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstNegative.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstAnd.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstGreaterThan.java
*
 /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstOr.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstString.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstFloatingPoint.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/NodeVisitor.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstInteger.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/util/ReflectionUtil.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/ELArithmetic.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/EvaluationContext.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstChoice.java

*
 /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstTrue.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstCompositeExpression.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/ExpressionFactoryImpl.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ArithmeticNode.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/ParseException.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/TokenMgrError.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/ELSupport.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/Node.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/lang/FunctionMapperImpl.java
*
 /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/util/MessageFactory.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstEmpty.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstGreaterThanEqual.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstDotSuffix.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstLessThanEqual.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstBracketSuffix.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/ValueExpressionImpl.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstMult.java
* /opt/cola/permits/175412180_1695324670.279257/0/javax-el-2-2-4-sources-jar/com/sun/el/parser/AstMod.java

1.103 jaxb-api 2.2.11

1.103.1 Available under license :

No license file was found, but licenses were detected in source scan.

* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlNs.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlValue.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlElements.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlID.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/XmlJavaTypeAdapters.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/JAXBIntrospector.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/JAXBElement.java

* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlType.java
*
 /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlElementDecl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/XmlAdapter.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/PropertyException.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlElementRef.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlEnum.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlTransient.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlSchema.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlIDREF.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/XmlJavaTypeAdapter.java
*
 /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlElement.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/PrintConversionEvent.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/ParseConversionEvent.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlRegistry.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlRootElement.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/NormalizedStringAdapter.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/CollapsedStringAdapter.java
*
 /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlElementRefs.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/HexBinaryAdapter.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlNsForm.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAttribute.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlEnumValue.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-

zip/javax/xml/bind/attachment/package.html
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/AbstractMarshallerImpl.java

* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/UnmarshallerHandler.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/MarshalException.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/AbstractUnmarshallerImpl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/DefaultValidationEventHandler.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/ValidationEventLocator.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/ParseConversionEventImpl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/NotIdentifiableEventImpl.java
*
 /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/util/Messages.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/NotIdentifiableEvent.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Element.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/JAXBException.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/ValidationEventLocatorImpl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/util/ValidationEventCollector.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/util/JAXBSource.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/TypeConstraintException.java
*
 /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/ContextFinder.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Marshaller.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Validator.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/Messages.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/ValidationEventImpl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/PrintConversionEventImpl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Unmarshaller.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/ValidationException.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/UnmarshalException.java
*

/opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/ValidationEvent.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/ValidationEventHandler.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/DatatypeConverter.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/JAXBContext.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/DatatypeConverterInterface.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/util/JAXBResult.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Messages.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/DataBindingException.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/GetPropertyAction.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlSeeAlso.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/JAXB.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/util/package.html
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/package.html
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/package.html
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/util/Messages.properties
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/helpers/Messages.properties
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Messages.properties
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/adapters/package.html
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/package.html
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlSchemaType.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/W3CDomHandler.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/DomHandler.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAnyElement.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlList.java

* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlMixed.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlElementWrapper.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAccessType.java
*
/opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/Binder.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAttachmentRef.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAnyAttribute.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlSchemaTypes.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlMimeType.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAccessorType.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/attachment/AttachmentUnmarshaller.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/SchemaOutputResolver.java
*
/opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAccessorOrder.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/attachment/AttachmentMarshaller.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlAccessOrder.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/annotation/XmlInlineBinaryData.java
No license file was found, but licenses were detected in source scan.

* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/JAXBPermission.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/DatatypeConverterImpl.java
* /opt/cola/permits/175413184_1695324670.404107/0/jaxb-api-2-2-11-sources-zip/javax/xml/bind/WhiteSpaceProcessor.java

1.104 jedis 2.8.1

1.104.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more contributor license
* agreements. See the NOTICE file distributed with this work for additional information regarding
* copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable
* law or agreed to in writing, software distributed under the License is distributed on an "AS IS"

Found in path(s):
* /opt/cola/permits/175419009_1695326855.803359/0/jedis-2-8-1-sources-jar/redis/clients/util/MurmurHash.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175419009_1695326855.803359/0/jedis-2-8-1-sources-jar/redis/clients/util/RedisInputStream.java

1.105 joda-time 2.7

1.105.1 Available under license :

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.

1.106 json-smart 1.3.1

1.106.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
*   Copyright 2011 JSON-SMART authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserBase.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/ContentHandler.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONStyleIdent.java
*
/opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserByteArray.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONObject.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONStyle.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/ContainerFactory.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserInputStream.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONArray.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONAwareEx.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-

jar/net/minidev/json/JSONNavi.java
*
/opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/ParseException.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONValue.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserMemory.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONStreamAwareEx.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/ContentHandlerCompressor.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserString.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONStreamAware.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParser.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONAware.java
*
/opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserReader.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/JSONParserStream.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/ContentHandlerDumy.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JStylerObj.java
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/parser/FakeContainerFactory.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/175429271_1695324682.5389547/0/json-smart-1-3-1-sources-jar/net/minidev/json/JSONUtil.java

1.107 jtoml 1.0.0

1.108 log4j-api 2.8

1.108.1 Available under license :

Apache Log4j API
Copyright 1999-2017 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

1.109 log4j-slf4j-impl 2.8

1.109.1 Available under license :

The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.110 httpcomponents-client 4.5.4

1.110.1 Available under license :

Apache HttpClient
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.111 metrics-health-checks 3.1.0

1.112 metrics-jetty 3.1.0

1.113 metrics-jvm 3.1.0

1.114 mongodb-driver 3.0.4

1.114.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBCollection.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoCollectionImpl.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/ListDatabasesIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ListIndexesIterableImpl.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/ListCollectionsIterable.java
*
 /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ListDatabasesIterableImpl.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/ListIndexesIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ListCollectionsIterableImpl.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBObjectCodec.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-

jar/com/mongodb/DBCollectionObjectFactory.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DB.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/CopyOnWriteMap.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/AbstractCopyOnWriteMap.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MapReduceIterableImpl.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-

jar/com/mongodb/AggregateIterableImpl.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/DistinctIterable.java
*
/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/AggregateIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/package-info.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/MapReduceIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DistinctIterableImpl.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/types/package-info.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/package-info.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/io/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008-2014 MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BSON.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBEncoderAdapter.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/ClassMapBasedObjectSerializer.java
*
/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/MongoIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ReplicaSetStatus.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-

jar/com/mongodb/BulkWriteOperation.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/RemoveRequest.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/LazyDBEncoder.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/QueryBuilder.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/FindIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoDatabaseImpl.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MapReduceOutput.java
*
/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBDecoderAdapter.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/Mongo.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BSONCallbackAdapter.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/ClassAncestry.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/AggregationOptions.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBCallbackFactory.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BSONTimestampCodec.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/LazyBSONCallback.java
*
/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/gridfs/GridFSFile.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/WriteConcernError.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MappingIterable.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/LazyDBObject.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/LazyDBCallback.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/ObjectSerializer.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/io/Bits.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBEncoderFactoryAdapter.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/ComputingMap.java

*

/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DefaultDBDecoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBAddress.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/ClassMap.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DefaultDBCallback.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/CommandResult.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkWriteHelper.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/JSON.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/types/CodeWScope.java

*

/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/QueryOperators.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/LazyDBList.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/AggregationOutput.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/types/BSONTimestamp.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoURI.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/InsertOptions.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBDecoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/gridfs/GridFS.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/gridfs/package-info.java

*

/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/LazyDBDecoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBRef.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/CompoundDBObjectCodec.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BasicBSONEncoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BSONObject.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-

jar/com/mongodb/client/MongoDatabase.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/EmptyBSONCallback.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBRefCodec.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBRefCodecProvider.java
*
 /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoCursorAdapter.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/package-info.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/UpdateRequest.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DocumentToDBRefTransformer.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/types/StringRangeSet.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BSONDecoder.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BSONCallback.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/UnacknowledgedBulkWriteResult.java
*
 /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BasicBSONObject.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/gridfs/GridFSInputFile.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBEncoderFactory.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkWriteException.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBObject.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBCursor.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ParallelScanOptions.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ReflectionDBObject.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkUpdateRequestBuilder.java
*
 /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoClientOptions.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BSONEncoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/gridfs/CLI.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBEncoder.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/LazyBSONList.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/WriteRequest.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BasicBSONCallback.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/ReplaceRequest.java
*
 /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/MongoCursor.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkWriteResult.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/package-info.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BasicDBObject.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkWriteError.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/LazyBSONDecoder.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoOptions.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/Bytes.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/package-info.java
*
 /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BasicDBObjectFactory.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBCallback.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/AcknowledgedBulkWriteResult.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/Cursor.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkWriteUpsert.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBDecoderFactory.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoClientURI.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/AbstractObjectSerializer.java

*

/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MapReduceCommand.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoClient.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/WriteResult.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BasicDBList.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/client/MongoCollection.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BasicDBObjectBuilder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/JSONSerializers.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBObjectCodecProvider.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/gridfs/GridFSDBFile.java

*

/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoBatchCursorAdapter.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DBObjects.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/OperationIterable.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/types/BasicBSONList.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/DefaultDBEncoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/InsertRequest.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/MongoMappingCursor.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/GroupCommand.java

*

/opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/BulkWriteRequestBuilder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/JSONParseException.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/BasicBSONDecoder.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/LazyBSONObject.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/org/bson/util/Function.java

* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-

jar/com/mongodb/DBObjectFactory.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/Util.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/util/JSONCallback.java
* /opt/cola/permits/175442163_1695324722.0818603/0/mongodb-driver-3-0-4-sources-jar/com/mongodb/FindIterableImpl.java

1.115 mongodb-driver-core 3.0.4

1.115.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/assertions/Assertions.java
No license file was found, but licenses were detected in source scan.

* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InternalStreamConnection.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Connection.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/AsyncConnection.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2015 MongoDB, Inc.
* Copyright 2012 The Netty Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/netty/ReadTimeoutHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008-2014 MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and

// make a special exception for a command with only a single item added to it. It's allowed to exceed maximum document size so that

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/BaseWriteCommandMessage.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/Projections.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/Filters.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/Sorts.java
No license file was found, but licenses were detected in source scan.

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DescriptionHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/ReadPreferenceServerSelector.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014-2015 MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/WriteError.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/CountOptions.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ExponentiallyWeightedMovingAverage.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/DeleteOneModel.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2005 Brian Goetz and Tim Peierls
* Released under the Creative Commons Attribution License
*   (http://creativecommons.org/licenses/by/2.5)
* Official home: http://www.jcip.net
*
* Any republication or derived work distributed in source code form
* must include this copyright and license notice.
*/

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/annotations/NotThreadSafe.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/annotations/ThreadSafe.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/annotations/Immutable.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2008-2014 MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceBatchCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceStatistics.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/Block.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/WriteModel.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/ServerSelector.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerConnectionState.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/AsyncConnectionSource.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AggregateExplainOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SaslAuthenticator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/ReadWriteBinding.java
*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Stream.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/MapReduceAction.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerId.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/CommandResultBaseCallback.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceToCollectionOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SocketSettings.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/management/ConnectionPoolStatisticsMBean.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoTimeoutException.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/OperationHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DeleteOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/NoOpConnectionListener.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/QueryProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionMessagesSentEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/SingleServerBinding.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/QueryResultCallback.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultClusterFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoSocketReadException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InternalStreamConnectionFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/QueryResult.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InternalConnectionInitializer.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AggregateToCollectionOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MixedBulkWriteOperation.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/async/AsyncBatchCursor.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/UpdateMessage.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/ExplainVerbosity.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/DuplicateKeyException.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterSettings.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/netty/NettyStream.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SocketStream.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/management/NullMBeanServer.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/UserExistsOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionPoolListener.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/authentication/NativeAuthenticationHelper.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/connection/ConcurrentPool.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoSocketReadTimeoutException.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/AbstractReferenceCounted.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/BaseCluster.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/InsertManyOptions.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InternalConnection.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoWaitQueueFullException.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ConnectionId.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/ReadPreference.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SocketStreamHelper.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DropCollectionOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-

jar/com/mongodb/operation/InsertOperation.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/LatencyMinimizingServerSelector.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoCursorNotFoundException.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/management/ConnectionPoolStatistics.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/diagnostics/logging/SLF4JLogger.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ChangeListener.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/netty/NettyByteBuf.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/FindAndModifyHelper.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/management/package-info.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/AsyncWriteBinding.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/InsertOneModel.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultServerConnection.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/package-info.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoQueryException.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/ParallelCollectionScanOptions.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/package-info.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ResponseBuffers.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InsertMessage.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/AsyncReadWriteBinding.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DistinctOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/netty/NettyStreamFactory.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/UserOperationHelper.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CommandResultDocumentCodec.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/BulkWriteOptions.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/X509Authenticator.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/WriteOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/TaggableReadPreference.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/ServerAddressSelector.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/InsertRequest.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoCredential.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DeleteProtocol.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/WriteConcernResult.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ReplyHeader.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/WriteBinding.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/UpdateCommandMessage.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterDescription.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/UpdateOptions.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/WriteResultCallback.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CommandResultCodecProvider.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/BufferProvider.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterableServerFactory.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/ConnectionString.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/UpdateCommandProtocol.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/validator/MappedFieldNameValidator.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/DeleteManyModel.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-

jar/com/mongodb/connection/GSSAPIAuthenticator.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/WriteCommandProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ChangeEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/ListDatabasesOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoBulkWriteException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AsyncOperationExecutor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/FindOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/diagnostics/logging/JULLogger.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceInlineResultsCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/CommandResultCallback.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InsertProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/Function.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DropDatabaseOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/StreamFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/NoOpClusterListener.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterFactory.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/NoOpConnectionPoolListener.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/FindAndReplaceOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoExecutionTimeoutException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionPoolListenerAdapter.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/result/DeleteResult.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/CreateCollectionOptions.java

*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/GenericWriteProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/GetMoreResultCallback.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/PlainAuthenticator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/CommandProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SingleServerCluster.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/connection/IndexMap.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/BulkWriteUpsert.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ScramSha1Authenticator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/AuthenticationMechanism.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/WriteConcernException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ByteBufferBsonOutput.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/KillCursorsMessage.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ProtocolHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionMessageReceivedEvent.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/BatchCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/BsonDocumentWrapperHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/ListCollectionsOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/PrimaryServerSelector.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/UpdateOneModel.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerMonitor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionPoolEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-

jar/com/mongodb/MongoSocketClosedException.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/KillCursorProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ConnectionPoolSettings.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ClusterEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ConnectionFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AsyncReadOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CommandWriteOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/RequestMessage.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/ParallelCollectionScanOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/ClusterBinding.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/annotations/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/FindOptions.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerSettings.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ClusterListenerAdapter.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/FutureAsyncCompletionHandler.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InternalConnectionFactory.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterType.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/async/ErrorHandlingResultCallback.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/connection/PowerOfTwoBufferPool.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/GroupOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceAsyncBatchCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Protocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoCommandException.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/AsyncClusterBinding.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DeleteCommandMessage.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerVersion.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/RenameCollectionOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/UpdateManyModel.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoSocketOpenException.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InsertCommandMessage.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SocketChannelStream.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceWithInlineResultsOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/ListIndexesOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/FindOperationHelper.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/ReadOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/async/SingleResultCallback.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/NativeAuthenticator.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoSocketWriteException.java

*

/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerType.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/UnacknowledgedWriteResultCallback.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CreateUserOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/FindAndUpdateOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoNamespace.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/AsynchronousSocketChannelStream.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/AsyncWritableByteChannel.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-

jar/com/mongodb/client/model/IndexOptions.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/diagnostics/logging/Logger.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionListener.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/BulkWriteResult.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/OperationExecutor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CommandResultArrayCodec.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ReplyMessage.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/management/JMXConnectionPoolListener.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ResponseCallback.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/validator/NoOpFieldNameValidator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/CommandHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/MapReduceInlineResultsAsyncCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoServerException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/OrderBy.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CursorHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/selector/CompositeServerSelector.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/diagnostics/logging/Loggers.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionEventMulticaster.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/AsyncReadBinding.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/UpdateProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/netty/NettyBufferProvider.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/validator/UpdateFieldNameValidator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DropUserOperation.java

*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerMonitorFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/ServerAddress.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Cluster.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DeleteMessage.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/ReferenceCounted.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultClusterableServerFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Server.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/management/MBeanServerFactory.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AggregateOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ConnectionDescription.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/UpdateRequest.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/QueryMessage.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CommandOperationHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/ConnectionSource.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/WriteRequest.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/UsageTrackingInternalConnection.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/ServerCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoSocketException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultServer.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Authenticator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/MessageSettings.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/FindAndDeleteOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-

jar/com/mongodb/management/MBeanServer.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/AsyncCompletionHandler.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionPoolOpenedEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/validator/CollectibleDocumentFieldNameValidator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionPoolWaitQueueEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/AsynchronousSocketChannelStreamFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ServerDescription.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/ReplaceOneModel.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DocumentHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ConnectionPool.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/UpdateOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CountOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/BulkWriteError.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/WriteCommandResultHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/DeleteRequest.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterableServer.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/WriteConcernError.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterConnectionMode.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/GetMoreMessage.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoSecurityException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/WriteProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/BaseWriteOperation.java

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultServerMonitorFactory.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SslSettings.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/MultiServerCluster.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/management/jmx/JMXMBeanServer.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AsyncQueryBatchCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ClusterListener.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/result/UpdateResult.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/package-info.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InternalStreamConnectionInitializer.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ClusterEventMulticaster.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionPoolEventMulticaster.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CommandReadOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/QueryBatchCursor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ClusterDescriptionChangedEvent.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/CommandMessage.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoInterruptedException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/GetMoreProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/connection/Pool.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/BulkWriteBatchCombiner.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DeleteCommandProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/ConnectionListenerAdapter.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ClusterId.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-

jar/com/mongodb/operation/CreateIndexesOperation.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoIncompatibleDriverException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoInternalException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/BaseQueryMessage.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/InsertCommandProtocol.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultConnectionFactory.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SocketStreamFactory.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/ExplainHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoClientException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/ReadBinding.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/ProtocolExecutor.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/SendMessageCallback.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/AsyncWriteOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/WriteConcern.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/CreateCollectionOperation.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/UpdateUserOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/RenameCollectionOptions.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/DropIndexOperation.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultServerMonitor.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 MongoDB, Inc.

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoWriteException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoNotPrimaryException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/binding/AbstractReferenceCounted.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/Tag.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoWriteConcernException.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/CursorType.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/TagSet.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultAuthenticator.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/MongoNodeIsRecoveringException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/DefaultConnectionPool.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/FindOneAndUpdateOptions.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/ErrorCategory.java
*
 /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/FindOneAndDeleteOptions.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/FindOneAndReplaceOptions.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/annotations/Beta.java
No license file was found, but licenses were detected in source scan.

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/
/**
 * Returns an array containing all of the elements in this deque, in
 * proper sequence (from first to last element).

 *
 * <p>The returned array will be "safe" in that no references to it are
 * maintained by this deque. (In other words, this method must allocate
 * a new array). The caller is thus free to modify the returned array.
 *
 * <p>This method acts as bridge between array-based and collection-based
 * APIs.
 *
 * @return an array containing all of the elements in this deque
 */

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/connection/ConcurrentLinkedDeque.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008-2014 MongoDB, Inc.
* Copyright 1999,2005 The Apache Software Foundation.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/Base64Codec.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/result/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/BsonArrayWriter.java
*
/opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/connection/netty/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/BsonArrayWrapper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/assertions/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/async/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/ReturnDocument.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/event/package-info.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/diagnostics/logging/package-info.java
No license file was found, but licenses were detected in source scan.

```
/*
```

```
* Copyright 2015 MongoDB, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/client/model/IndexModel.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/bulk/IndexRequest.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/internal/connection/SslHelper.java
* /opt/cola/permits/175442268_1695326916.39061/0/mongodb-driver-core-3-0-4-sources-jar/com/mongodb/operation/IndexHelper.java

1.116 stax 3.1.4

1.116.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/* StAX2 extension for StAX API (JSR-173).
*
* Copyright (c) 2005- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/Base64DecoderBase.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/util/StreamWriterDelegate.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/StringBase64Decoder.java
*
/opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/NumberUtil.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/CharArrayBase64Decoder.java
No license file was found, but licenses were detected in source scan.

/* Stax2 extension for basic Stax API (JSR-173).
*
* Copyright (c) 2005- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/typed/TypedXMLStreamReader.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/typed/Base64Variant.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/typed/Base64Variants.java
*
/opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/XMLStreamReader2.java
No license file was found, but licenses were detected in source scan.

/* Stax2 API extension for Streaming Api for Xml processing (StAX).
*
* Copyright (c) 2006- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in the file LICENSE which is
* included with the source code.
* You may not use this file except in compliance with the License.
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/dom/DOMWrappingWriter.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2Util.java
No license file was found, but licenses were detected in source scan.

/* Reference Implementation of
* Stax2 extension API (for basic Stax API, JSR-173)
*
* Copyright (c) 2005- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/ValueEncoderFactory.java
No license file was found, but licenses were detected in source scan.

/* Reference Implementation of
* Stax2 extension API (for basic Stax API, JSR-173)
*
* Copyright (c) 2008- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and

```
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/ValueDecoderFactory.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/typed/AsciiValueEncoder.java

No license file was found, but licenses were detected in source scan.

```
/* Stax2 API extension for Streaming Api for Xml processing (StAX).
*
* Copyright (c) 2006- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2WriterAdapter.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2ReaderImpl.java
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2WriterImpl.java

No license file was found, but licenses were detected in source scan.

```
/* Stax2 API extension for Streaming API for XML processing (StAX).
*
* Copyright (c) 2006- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in the file LICENSE which is
* included with the source code.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/dom/DOMWrappingReader.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175461681_1695324749.7461283/0/stax2-api-3-1-4-sources-jar/org/codehaus/stax2/ri/Stax2EventReaderImpl.java

1.117 tyrus-client 1.13

1.117.1 Available under license :

No license file was found, but licenses were detected in source scan.

* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/RetryAfterException.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/auth/BasicAuthenticator.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/RedirectException.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2014-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*

Found in path(s):
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/auth/AuthenticationException.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/auth/Authenticator.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/package-info.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/ClientProperties.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/auth/AuthConfig.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/ThreadPoolConfig.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/auth/Credentials.java
No license file was found, but licenses were detected in source scan.

* Copyright (c) 2013-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/TyrusClientEngine.java
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/auth/DigestAuthenticator.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*
* Copyright (c) 2011-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/ClientManager.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*

* Copyright (c) 2008-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/SslEngineConfigurator.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2007-2016 Oracle and/or its affiliates. All rights reserved.

```
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/175466462_1695326941.6953876/0/tyrus-client-1-13-sources-jar/org/glassfish/tyrus/client/SslContextConfigurator.java

1.118 tyrus-container-grizzly-client 1.13

1.118.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/package-info.java

* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/HttpCodecFilter.java
* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyClientProperties.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyClientContainer.java
* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyWriter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyClientSocket.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyClientFilter.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/TaskProcessor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-

jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyTransportTimeoutFilter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175466560_1695326944.7310467/0/tyrus-container-grizzly-client-1-13-sources-jar/org/glassfish/tyrus/container/grizzly/client/GrizzlyContainerProvider.java

1.119 tyrus-core 1.13

1.119.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/HandshakeException.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/BinaryFrame.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/OutputStreamToAsyncBinaryAdapter.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/WriterToAsyncTextAdapter.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/CharacterIterator.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/CloseFrame.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Handshake.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ServiceConfigurationError.java
*
 /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/UriParser.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Utf8DecodingException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/MultivaluedHashMap.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ComponentProvider.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ServiceFinder.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ComponentProviderService.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ReflectionHelper.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/TextFrame.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/TyrusFrame.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusEndpointWrapper.java
*
 /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/PathSegment.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/AbstractMultivaluedMap.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/MultivaluedStringMap.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/OsgiRegistry.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ProtocolHandler.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-

jar/org/glassfish/tyrus/core/StrictUtf8.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusWebSocket.java
*
 /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusWebSocketEngine.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusSession.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusRemoteEndpoint.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2004-2015 Oracle and/or its affiliates. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   - Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
*   - Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
*   - Neither the name of Oracle nor the names of its
*   contributors may be used to endorse or promote products derived
*   from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
 BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Base64Utils.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

*
* Copyright (c) 2013-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/CoderWrapper.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusFuture.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/NoOpByteArrayCoder.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-

jar/org/glassfish/tyrus/core/coder/ReaderDecoder.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/InputStreamDecoder.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/BufferedStringReader.java
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2011-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/NoOpByteBufferCoder.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/AnnotatedEndpoint.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/PrimitiveDecoders.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject

to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/PatternWithGroups.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/l10n/Localizable.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/UriTemplateParser.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/UriComponent.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/PathPattern.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/UriTemplate.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2010-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):

```
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/l10n/Localizer.java
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception as provided
 by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
```

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/DefaultComponentProvider.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/PrimitivesToWrappers.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/BufferedInputStream.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ExecutorServiceProvider.java
No license file was found, but licenses were detected in source scan.

* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/NoOpTextCoder.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/AsyncMessageHandler.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ProtocolException.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/WebSocketException.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/PingFrame.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusUpgradeResponse.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Masker.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/ToStringEncoder.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/PongFrame.java
*
/opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Version.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/SecKey.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/BasicMessageHandler.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/MessageTooBigException.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/BinaryBuffer.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/Frame.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/InputStreamBuffer.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/extension/ExtendedExtension.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-

jar/org/glassfish/tyrus/core/RequestContext.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/BaseContainer.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ReaderBuffer.java
*
 /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/AnnotatedClassValidityChecker.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/MessageHandlerManager.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Utils.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2014 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception as provided
 by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under

* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to
 such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/monitoring/MessageEventListener.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/BroadcastListener.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/CloseReasons.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/DefaultTyrusServerEndpointConfig.java
No license file was found, but licenses were detected in source scan.

# Copyright (c) 2012-2014 Oracle and/or its affiliates. All rights reserved.
# The contents of this file are subject to the terms of either the GNU
# General Public License Version 2 only ("GPL") or the Common Development
# and Distribution License("CDDL") (collectively, the "License"). You
# may not use this file except in compliance with the License. You can
# Oracle designates this particular file as subject to the "Classpath"
# exception as provided by Oracle in the GPL Version 2 section of the License

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/l10n/localization.properties
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2014-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.

*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/MaskingKeyGenerator.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/frame/package-info.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/package-info.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/package-info.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/monitoring/package-info.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/ClusterContext.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/SessionListener.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/DistributedSession.java
*
/opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/DebugContext.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/package-info.java

* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/monitoring/ApplicationEventListener.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/monitoring/EndpointEventListener.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/MaxSessions.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/RemoteSession.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/Beta.java
*
 /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusServerEndpointConfig.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/SessionEventListener.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/extension/package-info.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/cluster/package-info.java
No license file was found, but licenses were detected in source scan.

*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/l10n/LocalizableMessage.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/l10n/LocalizableMessageFactory.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/l10n/package-info.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/MultivaluedMap.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/internal/PathTemplate.java
No license file was found, but licenses were detected in source scan.

as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/coder/CoderAdapter.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/Match.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/uri/MatchComparator.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/ErrorCollector.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TyrusExtension.java
* /opt/cola/permits/175466658_1695324762.7455328/0/tyrus-core-1-13-sources-jar/org/glassfish/tyrus/core/TextBuffer.java

1.120 tyrus-spi 1.13

1.120.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU

```
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
/**
    * Open client socket - connect to endpoint specified with {@code url} parameter.
    * <p>
    * Called from ClientManager when {@link javax.websocket.WebSocketContainer#connectToServer(Class,
    * javax.websocket.ClientEndpointConfig, java.net.URI)} is invoked.
    *
    * @param cec          endpoint configuration. SPI consumer can access user properties, {@link
    *                     javax.websocket.ClientEndpointConfig.Configurator}, extensions and subprotocol
    *                     configuration,
    *                     etc..
    * @param properties   properties passed from client container. Don't mix up this with {@link
    *                     javax.websocket.ClientEndpointConfig#getUserProperties()}, these are Tyrus proprietary.
    * @param clientEngine one instance equals to one connection, cannot be reused. Implementation is expected to
```

```
call
   *                  {@link ClientEngine#createUpgradeRequest(ClientEngine.TimeoutHandler)}
 and {@link
   *                  ClientEngine#processResponse(UpgradeResponse, Writer,
   *                  org.glassfish.tyrus.spi.Connection.CloseListener)} (in that order).
   * @throws javax.websocket.DeploymentException when the client endpoint is invalid or when there is any other
(not
   *                                  specified) connection problem.
   * @throws java.io.IOException               when there is any I/O issue related to opening client socket or
   *                                  connecting to remote endpoint.
   */
```

Found in path(s):
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/ClientContainer.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2016 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]

Found in path(s):
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/Connection.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/ClientEngine.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/ReadHandler.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/UpgradeResponse.java
No license file was found, but licenses were detected in source scan.

* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013-2015 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"

Found in path(s):
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/CompletionHandler.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/WebSocketEngine.java
No license file was found, but licenses were detected in source scan.

* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/UpgradeRequest.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/ServerContainer.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/ServerContainerFactory.java
* /opt/cola/permits/175466756_1695324772.210957/0/tyrus-spi-1-13-sources-jar/org/glassfish/tyrus/spi/Writer.java

1.121 st4 4.0.8

1.121.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* [The "BSD license"]
* Copyright (c) 2011 Terence Parr
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR

```
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
grammar Group;

options {
 language=Java;
}

tokens {
 TRUE='true'; FALSE='false'; LBRACK='['; RBRACK=']';
}

@header {
/*
* [The "BSD license"]
*  Copyright (c) 2011 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the
 above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY
```

```
 OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
package org.stringtemplate.v4.compiler;

import java.util.Collections;
import java.util.Map;
import java.util.HashMap;
import java.util.List;
import java.util.ArrayList;
import org.stringtemplate.v4.misc.*;
import org.stringtemplate.v4.*;
import java.io.File;
}

@lexer::header {
/*
* [The "BSD license"]
*  Copyright (c) 2011 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the
 following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
package org.stringtemplate.v4.compiler;
import org.stringtemplate.v4.*;
import org.stringtemplate.v4.misc.*;
import java.io.File;
}

@members {
public STGroup group;

@Override
public void displayRecognitionError(String[] tokenNames,
                                    RecognitionException e)
{
   String msg = getErrorMessage(e, tokenNames);
   group.errMgr.groupSyntaxError(ErrorType.SYNTAX_ERROR, getSourceName(), e, msg);
}
@Override
public String getSourceName() {
   String fullFileName = super.getSourceName();
   File f = new File(fullFileName); // strip to simple name
   return f.getName();
}
public void error(String msg) {
   NoViableAltException e = new NoViableAltException("", 0, 0, input);
   group.errMgr.groupSyntaxError(ErrorType.SYNTAX_ERROR, getSourceName(), e, msg);
   recover(input, null);
}
}

@lexer::members {
public STGroup group;

@Override
public void reportError(RecognitionException e) {
   String msg = null;
   if ( e instanceof NoViableAltException
 ) {
       msg = "invalid character '"+(char)input.LA(1)+"'";
   }
   else if ( e instanceof MismatchedTokenException && ((MismatchedTokenException)e).expecting=='"' ) {
       msg = "unterminated string";
   }
   else {
       msg = getErrorMessage(e, getTokenNames());
   }
   group.errMgr.groupSyntaxError(ErrorType.SYNTAX_ERROR, getSourceName(), e, msg);
}
@Override
public String getSourceName() {
   String fullFileName = super.getSourceName();
   File f = new File(fullFileName); // strip to simple name
   return f.getName();
}
}

group[STGroup group, String prefix]
@init {
GroupLexer lexer = (GroupLexer)input.getTokenSource();
this.group = lexer.group = $group;
}
 : oldStyleHeader?
  delimiters?
     ( 'import' STRING {group.importTemplates($STRING);}
  | 'import' // common error: name not in string
   {
   MismatchedTokenException e = new MismatchedTokenException(STRING, input);
   reportError(e);
   }
   ID ('.' ID)* // might be a.b.c.d
  )*
  def[prefix]*
  EOF
 ;

oldStyleHeader
 // ignore but lets us use this parser in AW for both v3 and v4
   :   'group' ID ( ':' ID )?
     ( 'implements' ID (',' ID)* )?
     ';'
 ;

groupName returns [String name]
@init {StringBuilder buf = new StringBuilder();}
 : a=ID {buf.append($a.text);} ('.' a=ID {buf.append($a.text);})*
 ;

delimiters
   : 'delimiters' a=STRING ',' b=STRING
     {
     group.delimiterStartChar=$a.getText().charAt(1);
       group.delimiterStopChar=$b.getText().charAt(1);
       }
   ;

/** Match template and dictionary defs outside of (...)+ loop in group.
*  The key is catching while still in the loop; must keep prediction of
*  elements separate from "stay in loop" prediction.
*/
def[String prefix] : templateDef[prefix] | dictDef ;
```

```
 catch[RecognitionException re] {
  // pretend we already saw an error here
  state.lastErrorIndex = input.index();
  error("garbled template definition starting at '"+input.LT(1).getText()+"'");
 }

templateDef[String prefix]
@init {
   String
 template=null;
   int n=0; // num char to strip from left, right of template def
}
 : ( '@' enclosing=ID '.' name=ID '(' ')'
  | name=ID '(' formalArgs ')'
  )
     '::='
     {Token templateToken = input.LT(1);}
     ( STRING     {template=$STRING.text; n=1;}
     | BIGSTRING  {template=$BIGSTRING.text; n=2;}
     | BIGSTRING_NO_NL  {template=$BIGSTRING_NO_NL.text; n=2;}
     | {
      template = "";
      String msg = "missing template at '"+input.LT(1).getText()+"'";
           NoViableAltException e = new NoViableAltException("", 0, 0, input);
        group.errMgr.groupSyntaxError(ErrorType.SYNTAX_ERROR, getSourceName(), e, msg);
        }
     )
     {
     if ( $name.index >= 0 ) { // if ID missing
   template = Misc.strip(template, n);
   String templateName = $name.text;
   if ( prefix.length()>0 ) templateName = prefix+$name.text;
   String enclosingTemplateName = $enclosing.text;
   if (enclosingTemplateName != null && enclosingTemplateName.length()>0 && prefix.length()>0)
 {
    enclosingTemplateName = prefix + enclosingTemplateName;
    }
   group.defineTemplateOrRegion(templateName, enclosingTemplateName, templateToken,
           template, $name, $formalArgs.args);
  }
     }
 |   alias=ID '::=' target=ID  {group.defineTemplateAlias($alias, $target);}
 ;

formalArgs returns[List<FormalArgument> args = new ArrayList<FormalArgument>()]
scope {
 boolean hasOptionalParameter;
}
```

```
@init { $formalArgs::hasOptionalParameter = false; }
 : formalArg[$args] (',' formalArg[$args])*
 |
 ;

formalArg[List<FormalArgument> args]
 : ID
  ( '=' a=(STRING|ANONYMOUS_TEMPLATE|'true'|'false') {$formalArgs::hasOptionalParameter = true;}
  | '=' a='[' ']' {$formalArgs::hasOptionalParameter = true;}
  | {
   if ($formalArgs::hasOptionalParameter) {
    group.errMgr.compileTimeError(ErrorType.REQUIRED_PARAMETER_AFTER_OPTIONAL,
             null, $ID);
   }
   }
  )
  {$args.add(new FormalArgument($ID.text, $a));}
   ;

/*
suffix returns [int cardinality=FormalArgument.REQUIRED]
    :   OPTIONAL
   |  STAR
   |  PLUS
 |
   ;
       */

dictDef
 : ID '::=' dict
       {
       if ( group.rawGetDictionary($ID.text)!=null ) {
   group.errMgr.compileTimeError(ErrorType.MAP_REDEFINITION, null, $ID);
       }
       else if ( group.rawGetTemplate($ID.text)!=null ) {
   group.errMgr.compileTimeError(ErrorType.TEMPLATE_REDEFINITION_AS_MAP, null, $ID);
       }
       else {
           group.defineDictionary($ID.text, $dict.mapping);
       }
       }
 ;

dict returns [Map<String,Object> mapping]
@init {mapping=new HashMap<String,Object>();}
 :   '[' dictPairs[mapping] ']'
 ;
```

```
dictPairs[Map<String,Object> mapping]
   : keyValuePair[mapping]
    (',' keyValuePair[mapping])* (',' defaultValuePair[mapping])?
   | defaultValuePair[mapping]
   ;
 catch[RecognitionException re] {
  error("missing dictionary entry at '"+input.LT(1).getText()+"'");
 }

defaultValuePair[Map<String,Object> mapping]
 : 'default' ':' keyValue {mapping.put(STGroup.DEFAULT_KEY,
 $keyValue.value);}
 ;

keyValuePair[Map<String,Object> mapping]
 : STRING ':' keyValue {mapping.put(Misc.replaceEscapes(Misc.strip($STRING.text, 1)), $keyValue.value);}
 ;

keyValue returns [Object value]
 : BIGSTRING   {$value = group.createSingleton($BIGSTRING);}
 | BIGSTRING_NO_NL  {$value = group.createSingleton($BIGSTRING_NO_NL);}
 | ANONYMOUS_TEMPLATE {$value = group.createSingleton($ANONYMOUS_TEMPLATE);}
 | STRING    {$value = Misc.replaceEscapes(Misc.strip($STRING.text, 1));}
 | TRUE    {$value = true;}
 | FALSE    {$value = false;}
 | '[' ']'    {$value = Collections.emptyList();}
 | {input.LT(1).getText().equals("key")}?=> ID
      {$value = STGroup.DICT_KEY;}
 ;
 catch[RecognitionException re] {
  error("missing value for key at '"+input.LT(1).getText()+"'");
 }

ID : ('a'..'z'|'A'..'Z'|'_') ('a'..'z'|'A'..'Z'|'0'..'9'|'-'|'_')*
 ;

STRING
 : '"'
  ( '\\' '"'
  | '\\' ~'"'
  | {
   String msg = "\\n in string";
     NoViableAltException e = new NoViableAltException("",
 0, 0, input);
   group.errMgr.groupLexerError(ErrorType.SYNTAX_ERROR, getSourceName(), e, msg);
   }
   '\n'
  | ~('\\'|'"'|'\n')
 )*
 '"'
      {
      String txt = getText().replaceAll("\\\\\"","\"");
 setText(txt);
 }
;

BIGSTRING_NO_NL // same as BIGSTRING but means ignore newlines later
 : '<%' ( . )* '%>'
      // %\> is the escape to avoid end of string
      {
      String txt = getText().replaceAll("\%\\\\>","\%>");
 setText(txt);
 }
;

/** Match <<...>> but also allow <<..<x>>> so we can have tag on end.
   Escapes: >\> means >> inside of <<...>>.
   Escapes: \>> means >> inside of <<...>> unless at end like <<...\>>>>.
   In that case, use <%..>>%> instead.
*/
BIGSTRING
 : '<<'
  ( options {greedy=false;}
  : '\\' '>'  // \> escape
  | '\\' ~'>' // allow this but don't collapse in action
  | ~'\\'
  )*
      '>>'
      {
      String txt = getText();
      txt = Misc.replaceEscapedRightAngle(txt); // replace \> with > unless <\\>
  setText(txt);
  }
 ;

ANONYMOUS_TEMPLATE
    : '{'
    {
  Token templateToken = new CommonToken(input, ANONYMOUS_TEMPLATE, 0, getCharIndex(),
getCharIndex());
  STLexer lexer =
   new STLexer(group.errMgr, input, templateToken, group.delimiterStartChar, group.delimiterStopChar);
  lexer.subtemplateDepth = 1;
  Token t = lexer.nextToken();
  while ( lexer.subtemplateDepth>=1 || t.getType()!=STLexer.RCURLY ) {
   if ( t.getType()==STLexer.EOF_TYPE ) {
        MismatchedTokenException e = new MismatchedTokenException('}', input);
    String msg = "missing final '}' in {...} anonymous template";
      group.errMgr.groupLexerError(ErrorType.SYNTAX_ERROR, getSourceName(), e, msg);
    break;
   }
   t = lexer.nextToken();
  }
  }
    // don't match '}' here; our little {...} scanner loop matches it
    // to terminate.
   ;

COMMENT
   :   '/*' ( options {greedy=false;} : . )* '*/' {skip();}
   ;

LINE_COMMENT
   : '//' ~('\n'|'\r')* '\r'? '\n' {skip();}
   ;

WS  : (' '|'\r'|'\t'|'\n')
 {skip();} ;
```

Found in path(s):
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/Group.g
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2011 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY
```

```
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

tree grammar CodeGenerator;

options {
 language=Java;
 tokenVocab=STParser;
 ASTLabelType=CommonTree;
}

@header {
/*
* [The "BSD license"]
*  Copyright (c) 2011 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above
 copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY
 OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

```
package org.stringtemplate.v4.compiler;
import org.stringtemplate.v4.misc.*;
import org.stringtemplate.v4.*;
}

@members {
 String outermostTemplateName; // name of overall template
 CompiledST outermostImpl;
 Token templateToken;   // overall template token
 String template;      // overall template text
 ErrorManager errMgr;
 public CodeGenerator(TreeNodeStream input, ErrorManager errMgr, String name, String template, Token
templateToken) {
  this(input, new RecognizerSharedState());
  this.errMgr = errMgr;
  this.outermostTemplateName = name;
  this.template = template;
  this.templateToken = templateToken;
 }

 // convience funcs to hide offensive sending of emit messages to
 // CompilationState temp data object.

 public void emit1(CommonTree opAST, short opcode, int arg)
 {
  $template::state.emit1(opAST, opcode, arg);
 }
 public void emit1(CommonTree opAST, short opcode, String arg) {
  $template::state.emit1(opAST, opcode, arg);
 }
 public void emit2(CommonTree opAST, short opcode, int arg, int arg2) {
  $template::state.emit2(opAST, opcode, arg, arg2);
 }
 public void emit2(CommonTree opAST, short opcode, String s, int arg2) {
  $template::state.emit2(opAST, opcode, s, arg2);
 }
   public void emit(CommonTree opAST, short opcode) {
  $template::state.emit(opAST, opcode);
 }
 public void insert(int addr, short opcode, String s) {
  $template::state.insert(addr, opcode, s);
 }
 public void setOption(CommonTree id) {
  $template::state.setOption(id);
 }
 public void write(int addr, short value) {
  $template::state.write(addr,value);
 }
 public int address() { return $template::state.ip; }
 public void func(CommonTree id) { $template::state.func(templateToken, id); }
 public void refAttr(CommonTree id) { $template::state.refAttr(templateToken, id); }
 public
 int defineString(String s) { return $template::state.defineString(s); }
}

templateAndEOF : template[null,null] EOF; // hush warning; ignore

template[String name, List<FormalArgument> args] returns [CompiledST impl]
scope {
   CompilationState state; // automatically get a new state pointer per invocation
}
@init {
 $template::state = new CompilationState(errMgr, name, input.getTokenStream());
 $impl = $template::state.impl;
 if ( $template.size() == 1 ) outermostImpl = $impl;
 $impl.defineFormalArgs($args); // make sure args are defined prior to compilation
 if ( name!=null && name.startsWith(Compiler.SUBTEMPLATE_PREFIX) ) {
     $impl.addArg(new FormalArgument("i"));
     $impl.addArg(new FormalArgument("i0"));
   }
 $impl.template = template; // always forget the entire template; char indexes are relative to it
}
 : chunk
  { // finish off the CompiledST result
       if ( $template::state.stringtable!=null ) $impl.strings = $template::state.stringtable.toArray();

     $impl.codeSize = $template::state.ip;
  }
 ;

chunk
 : element*
 ;

element
 : ^(INDENTED_EXPR INDENT compoundElement[$INDENT]) // ignore indent in front of IF and region blocks
 | compoundElement[null]
 | ^(INDENTED_EXPR INDENT {$template::state.indent($INDENT);} singleElement
{$template::state.emit(Bytecode.INSTR_DEDENT);})
 | singleElement
 ;

singleElement
 : exprElement
 | TEXT
  {
```

```
  if ( $TEXT.text.length()>0 ) {
   emit1($TEXT,Bytecode.INSTR_WRITE_STR, $TEXT.text);
  }
  }

 | NEWLINE {emit($NEWLINE, Bytecode.INSTR_NEWLINE);}
 ;

compoundElement[CommonTree indent]
 : ifstat[indent]
 | region[indent]
 ;

exprElement
@init { short op = Bytecode.INSTR_WRITE; }
 : ^( EXPR expr (exprOptions {op=Bytecode.INSTR_WRITE_OPT;})? )
  {
  /*
  CompilationState state = $template::state;
  CompiledST impl = state.impl;
  if ( impl.instrs[state.ip-1] == Bytecode.INSTR_LOAD_LOCAL ) {
   impl.instrs[state.ip-1] = Bytecode.INSTR_WRITE_LOCAL;
  }
  else {
   emit($EXPR, op);
  }
  */
  emit($EXPR,
 op);
  }
 ;

region[CommonTree indent] returns [String name]
@init {
 if ( indent!=null ) $template::state.indent(indent);
}
@after {
 if ( indent!=null ) $template::state.emit(Bytecode.INSTR_DEDENT);
}
 : ^( REGION ID
   {$name = STGroup.getMangledRegionName(outermostTemplateName, $ID.text);}
   template[$name,null]
   {
   CompiledST sub = $template.impl;
         sub.isRegion = true;
         sub.regionDefType = ST.RegionType.EMBEDDED;
         sub.templateDefStartToken = $ID.token;
   //sub.dump();
   outermostImpl.addImplicitlyDefinedTemplate(sub);
   emit2($start, Bytecode.INSTR_NEW, $region.name, 0);
   emit($start, Bytecode.INSTR_WRITE);
   }
   )
 ;

subtemplate returns [String name, int nargs]
@init {
   $name = Compiler.getNewSubtemplateName();
 List<FormalArgument> args = new ArrayList<FormalArgument>();
}
 : ^( SUBTEMPLATE
   (^(ARGS (ID {args.add(new FormalArgument($ID.text));})+))*
   {$nargs = args.size();}
   template[$name,args]
   {
   CompiledST sub = $template.impl;
   sub.isAnonSubtemplate
 = true;
         sub.templateDefStartToken = $SUBTEMPLATE.token;
           sub.ast = $SUBTEMPLATE;
           sub.ast.setUnknownTokenBoundaries();
           sub.tokens = input.getTokenStream();
   //sub.dump();
   outermostImpl.addImplicitlyDefinedTemplate(sub);
   }
   )
 | SUBTEMPLATE // {}
   {
   CompiledST sub = new CompiledST();
   sub.name = $name;
   sub.template = "";
   sub.addArg(new FormalArgument("i"));
   sub.addArg(new FormalArgument("i0"));
   sub.isAnonSubtemplate = true;
         sub.templateDefStartToken = $SUBTEMPLATE.token;
           sub.ast = $SUBTEMPLATE;
           sub.ast.setUnknownTokenBoundaries();
           sub.tokens = input.getTokenStream();
   //sub.dump();
   outermostImpl.addImplicitlyDefinedTemplate(sub);
   }
 ;

ifstat[CommonTree indent]
@init {
   /** Tracks address of branch operand (in code block).  It's how
```

```
    *  we backpatch forward references when generating code for IFs.
    */
   int prevBranchOperand
 = -1;
   /** Branch instruction operands that are forward refs to end of IF.
    *  We need to update them once we see the endif.
    */
   List<Integer> endRefs = new ArrayList<Integer>();
   if ( indent!=null ) $template::state.indent(indent);
}
@after {
 if ( indent!=null ) $template::state.emit(Bytecode.INSTR_DEDENT);
}
 : ^( i='if' conditional
   {
         prevBranchOperand = address()+1;
         emit1($i,Bytecode.INSTR_BRF, -1); // write placeholder as branch target
   }
   chunk
   ( ^(eif='elseif'
    {
    endRefs.add(address()+1);
    emit1($eif,Bytecode.INSTR_BR, -1); // br end
    // update previous branch instruction
    write(prevBranchOperand, (short)address());
    prevBranchOperand = -1;
    }
    ec=conditional
    {
          prevBranchOperand = address()+1;
          // write placeholder as branch target
          emit1($ec.start, Bytecode.INSTR_BRF, -1);
    }
    chunk
    )
   )*
   ( ^( el='else'
     {
     endRefs.add(address()+1);
     emit1($el,
 Bytecode.INSTR_BR, -1); // br end
     // update previous branch instruction
     write(prevBranchOperand, (short)address());
     prevBranchOperand = -1;
     }
     chunk
     )
   )?
```

```
  )
  {
  if ( prevBranchOperand>=0 ) {
   write(prevBranchOperand, (short)address());
  }
      for (int opnd : endRefs) write(opnd, (short)address());
  }
 ;

conditional
 : ^(OR conditional conditional)  {emit($OR, Bytecode.INSTR_OR);}
 | ^(AND conditional conditional)  {emit($AND, Bytecode.INSTR_AND);}
 | ^(BANG conditional)     {emit($BANG, Bytecode.INSTR_NOT);}
 | expr // not all expr are valid, but reuse code gen (parser restricts syntax)
 ;

exprOptions : {emit($start, Bytecode.INSTR_OPTIONS);} ^(OPTIONS option*) ;

option : ^('=' ID expr) {setOption($ID);} ;

expr
@init {int nt = 0, ne = 0;}
 : ^(ZIP ^(ELEMENTS (expr {ne++;})+) mapTemplateRef[ne])
  {emit1($ZIP, Bytecode.INSTR_ZIP_MAP, ne);}
 | ^(MAP expr (mapTemplateRef[1] {nt++;})+)
  {
  if ( nt>1 ) emit1($MAP, nt>1?Bytecode.INSTR_ROT_MAP:Bytecode.INSTR_MAP,
 nt);
  else emit($MAP, Bytecode.INSTR_MAP);
  }
 | prop
 | includeExpr
 ;

prop: ^(PROP expr ID)      {emit1($PROP, Bytecode.INSTR_LOAD_PROP, $ID.text);}
 | ^(PROP_IND expr expr)    {emit($PROP_IND, Bytecode.INSTR_LOAD_PROP_IND);}
 ;

mapTemplateRef[int num_exprs]
 : ^( INCLUDE ID
   {for (int i=1; i<=$num_exprs; i++) emit($INCLUDE,Bytecode.INSTR_NULL);}
   args
  )
  {
  if ( $args.passThru ) emit1($start, Bytecode.INSTR_PASSTHRU, $ID.text);
  if ( $args.namedArgs ) emit1($INCLUDE, Bytecode.INSTR_NEW_BOX_ARGS, $ID.text);
  else emit2($INCLUDE, Bytecode.INSTR_NEW, $ID.text, $args.n+$num_exprs);
  }
 | subtemplate
  {
  if ( $subtemplate.nargs != $num_exprs ) {
          errMgr.compileTimeError(ErrorType.ANON_ARGUMENT_MISMATCH,
                templateToken, $subtemplate.start.token, $subtemplate.nargs, $num_exprs);
  }
  for (int i=1; i<=$num_exprs; i++) emit($subtemplate.start,Bytecode.INSTR_NULL);
       emit2($subtemplate.start, Bytecode.INSTR_NEW,

         $subtemplate.name,
               $num_exprs);
  }

 | ^( INCLUDE_IND expr
   {
   emit($INCLUDE_IND,Bytecode.INSTR_TOSTR);
   for (int i=1; i<=$num_exprs; i++) emit($INCLUDE_IND,Bytecode.INSTR_NULL);
   }
   args
   {
   emit1($INCLUDE_IND, Bytecode.INSTR_NEW_IND, $args.n+$num_exprs);
   }
   )
 ;

includeExpr
 : ^(EXEC_FUNC ID expr?)  {func($ID);}
 | ^(INCLUDE ID args)
  {
  if ( $args.passThru ) emit1($start, Bytecode.INSTR_PASSTHRU, $ID.text);
  if ( $args.namedArgs ) emit1($INCLUDE, Bytecode.INSTR_NEW_BOX_ARGS, $ID.text);
  else emit2($INCLUDE, Bytecode.INSTR_NEW, $ID.text, $args.n);
  }
 | ^(INCLUDE_SUPER ID args)
  {
  if ( $args.passThru ) emit1($start, Bytecode.INSTR_PASSTHRU, $ID.text);
  if ( $args.namedArgs ) emit1($INCLUDE_SUPER, Bytecode.INSTR_SUPER_NEW_BOX_ARGS, $ID.text);
  else emit2($INCLUDE_SUPER, Bytecode.INSTR_SUPER_NEW, $ID.text, $args.n);
  }
 | ^(INCLUDE_REGION ID)  {
         CompiledST impl =
          Compiler.defineBlankRegion(outermostImpl, $ID.token);
         //impl.dump();
         emit2($INCLUDE_REGION,Bytecode.INSTR_NEW,impl.name,0);
         }
 | ^(INCLUDE_SUPER_REGION
 ID) {
                           String mangled =
                              STGroup.getMangledRegionName(outermostImpl.name, $ID.text);
        emit2($INCLUDE_SUPER_REGION,Bytecode.INSTR_SUPER_NEW,mangled,0);
        }
 | primary
 ;

primary
 : ID    {refAttr($ID);}
 | STRING   {emit1($STRING,Bytecode.INSTR_LOAD_STR, Misc.strip($STRING.text,1));}
 | TRUE   {emit($TRUE, Bytecode.INSTR_TRUE);}
 | FALSE   {emit($FALSE, Bytecode.INSTR_FALSE);}
 | subtemplate  // push a subtemplate but ignore args since we can't pass any to it here
                {emit2($start,Bytecode.INSTR_NEW, $subtemplate.name, 0);}
 | list
 | ^( INCLUDE_IND
   expr   {emit($INCLUDE_IND, Bytecode.INSTR_TOSTR);}
   args        {emit1($INCLUDE_IND, Bytecode.INSTR_NEW_IND, $args.n);}
   )
 | ^(TO_STR expr) {emit($TO_STR, Bytecode.INSTR_TOSTR);}
 ;

arg : expr ;

args returns [int
 n=0, boolean namedArgs=false, boolean passThru]
 : ( arg {$n++;} )+
 | {emit($args.start, Bytecode.INSTR_ARGS); $namedArgs=true;}
  ( ^(eq='=' ID expr)
   {$n++; emit1($eq, Bytecode.INSTR_STORE_ARG, defineString($ID.text));}
  )+
  ( '...' {$passThru=true;} )?
  |   '...' {$passThru=true; emit($args.start, Bytecode.INSTR_ARGS); $namedArgs=true;}
 |
 ;

list: {emit($start, Bytecode.INSTR_LIST);}
  ^(LIST (listElement {emit($listElement.start, Bytecode.INSTR_ADD);})* )
 ;

listElement : expr | NULL {emit($NULL,Bytecode.INSTR_NULL);} ;
```

Found in path(s):
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/CodeGenerator.g
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2012 Terence Parr
*  Copyright (c) 2012 Sam Harwell
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 IMPLIED WARRANTIES
*  OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
*  THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/AmbiguousMatchException.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/TypeRegistry.java
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2011 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
```

```
/** Build an AST from a single StringTemplate template */
parser grammar STParser;

options {
 language=Java;
 tokenVocab=STLexer;
 TokenLabelType=CommonToken;
 output=AST;
 ASTLabelType=CommonTree;
}

tokens {
 EXPR; OPTIONS; PROP; PROP_IND; INCLUDE; INCLUDE_IND; EXEC_FUNC; INCLUDE_SUPER;
 INCLUDE_SUPER_REGION; INCLUDE_REGION; TO_STR; LIST; MAP; ZIP; SUBTEMPLATE; ARGS;
 ELEMENTS; REGION; NULL; INDENTED_EXPR;
 }

@header
 {
package org.stringtemplate.v4.compiler;
import org.stringtemplate.v4.misc.ErrorManager;
import org.stringtemplate.v4.misc.ErrorType;
}

@members {
ErrorManager errMgr;
Token templateToken;
public STParser(TokenStream input, ErrorManager errMgr, Token templateToken) {
 this(input);
 this.errMgr = errMgr;
 this.templateToken = templateToken;
}
@Override
protected Object recoverFromMismatchedToken(IntStream input, int ttype, BitSet follow)
 throws RecognitionException
{
 throw new MismatchedTokenException(ttype, input);
}
}

@rulecatch {
  catch (RecognitionException re) { throw re; }
}

templateAndEOF : template EOF -> template? ;

template : element* ;

element
 : {input.LT(1).getCharPositionInLine()==0}? INDENT? COMMENT NEWLINE -> // throw away
 | INDENT singleElement -> ^(INDENTED_EXPR INDENT singleElement?) // singleElement is optional to handle
error returning nil
 | singleElement
 | compoundElement
 ;

singleElement
 : exprTag
 | TEXT
 | NEWLINE
 | COMMENT! // throw away
 ;

compoundElement
 : ifstat
 | region
 ;

exprTag
 : LDELIM
 expr ( ';' exprOptions )? RDELIM
  -> ^(EXPR[$LDELIM,"EXPR"] expr exprOptions?)
 ;

region
@init {Token indent=null;}
 : i=INDENT? x=LDELIM '@' ID RDELIM {if (input.LA(1)!=NEWLINE) indent=$i;}
  template
  INDENT? LDELIM '@end' RDELIM
  // kill \n for <@end> on line by itself if multi-line embedded region
  ({$region.start.getLine()!=input.LT(1).getLine()}?=> NEWLINE)?
  -> {indent!=null}?
     ^(INDENTED_EXPR $i ^(REGION[$x] ID template?))
  ->                    ^(REGION[$x] ID template?)
 ;

subtemplate
 : lc='{' (ids+= ID ( ',' ids+= ID )* '|' )? template INDENT? '}'
  // ignore final INDENT before } as it's not part of outer indent
  -> ^(SUBTEMPLATE[$lc,"SUBTEMPLATE"] ^(ARGS $ids)* template?)
 ;

ifstat
@init {Token indent=null;}
 : i=INDENT? LDELIM 'if' '(' c1=conditional ')' RDELIM {if (input.LA(1)!=NEWLINE) indent=$i;}
   t1=template
   ( INDENT? LDELIM 'elseif' '(' c2+=conditional ')' RDELIM t2+=template )*
   (
 INDENT? LDELIM 'else' RDELIM t3=template )?
   INDENT? endif= LDELIM 'endif'
  RDELIM
  // kill \n for <endif> on line by itself if multi-line IF
  ({$ifstat.start.getLine()!=input.LT(1).getLine()}?=> NEWLINE)?
  -> {indent!=null}?
     ^(INDENTED_EXPR $i ^('if' $c1 $t1? ^('elseif' $c2 $t2)* ^('else' $t3?)?))
  ->                    ^('if' $c1 $t1? ^('elseif' $c2 $t2)* ^('else' $t3?)?)
 ;

conditional
scope {
 boolean inside;
}
 : andConditional ( '||'^ andConditional )*
 ;

andConditional : notConditional ( '&&'^ notConditional )* ;

notConditional
 : '!'^ notConditional
 | memberExpr
 ;

notConditionalExpr
 : (ID->ID)
  ( p='.' prop=ID       -> ^(PROP[$p,"PROP"] $notConditionalExpr $prop)
  | p='.' '(' mapExpr ')'    -> ^(PROP_IND[$p,"PROP_IND"] $notConditionalExpr mapExpr)
  )*
 ;
```

```
exprOptions : option ( ',' option )* -> ^(OPTIONS option*) ;

option
@init {
 String id = input.LT(1).getText();
 String defVal = Compiler.defaultOptionValues.get(id);
 boolean validOption = Compiler.supportedOptions.get(id)!=null;
}
 : ID
  {
  if
 ( !validOption ) {
          errMgr.compileTimeError(ErrorType.NO_SUCH_OPTION, templateToken, $ID, $ID.text);
  }
  }
  ( '=' exprNoComma      -> {validOption}? ^('=' ID exprNoComma)
          ->
  | {
   if ( defVal==null ) {
    errMgr.compileTimeError(ErrorType.NO_DEFAULT_VALUE, templateToken, $ID);
   }
   }
          -> {validOption&&defVal!=null}?
             ^(EQUALS["="] ID STRING[$ID,'"'+defVal+'"'])
          ->
  )
 ;

exprNoComma
 : memberExpr
  ( ':' mapTemplateRef     -> ^(MAP memberExpr mapTemplateRef)
  |          -> memberExpr
  )
 ;

expr : mapExpr ;

// more complicated than necessary to avoid backtracking, which ruins
// error handling
mapExpr
 : memberExpr
  ( (c=',' memberExpr)+ col=':' mapTemplateRef
          -> ^(ZIP[$col] ^(ELEMENTS memberExpr+) mapTemplateRef)
  |          -> memberExpr
  )
  ( {if ($x!=null) $x.clear();} // don't keep queueing x; new list for each iteration
   col=':' x+=mapTemplateRef
 ({$c==null}?=> ',' x+=mapTemplateRef )*
           -> ^(MAP[$col] $mapExpr $x+)
  )*
 ;

/**
expr:template(args)  apply template to expr
expr:{arg | ...}     apply subtemplate to expr
expr:(e)(args)       convert e to a string template name and apply to expr
*/
mapTemplateRef
 : ID '(' args ')'       -> ^(INCLUDE ID args?)
 | subtemplate
 | lp='(' mapExpr rp=')' '(' argExprList? ')' -> ^(INCLUDE_IND mapExpr argExprList?)
 ;

memberExpr
 : (includeExpr->includeExpr)
  ( p='.' ID       -> ^(PROP[$p,"PROP"] $memberExpr ID)
  | p='.' '(' mapExpr ')'    -> ^(PROP_IND[$p,"PROP_IND"] $memberExpr mapExpr)
  )*
 ;

includeExpr
options {k=2;} // prevent full LL(*), which fails, falling back on k=1; need k=2
 : {Compiler.funcs.containsKey(input.LT(1).getText())}? // predefined function
  ID '(' expr? ')'      -> ^(EXEC_FUNC ID expr?)
 | 'super' '.' ID '(' args ')'    -> ^(INCLUDE_SUPER ID args?)
 | ID '(' args ')'       -> ^(INCLUDE ID args?)
 | '@' 'super' '.' ID '(' rp=')'   -> ^(INCLUDE_SUPER_REGION
 ID)
 | '@' ID '(' rp=')'      -> ^(INCLUDE_REGION ID)
 | primary
 ;

primary
 : ID
 | STRING
 | TRUE
 | FALSE
 | subtemplate
 | list
 | {$conditional.size()>0}?=>  '('! conditional ')'!
 | {$conditional.size()==0}?=> lp='(' expr ')'
  ( '(' argExprList? ')'          -> ^(INCLUDE_IND[$lp] expr argExprList?)
  |         -> ^(TO_STR[$lp] expr)
  )
 ;
```

```
args: argExprList
 | namedArg ( ',' namedArg )* (',' '...')? -> namedArg+ '...'?
   |   '...'
 |
 ;

argExprList : arg ( ',' arg )* -> arg+ ;

arg : exprNoComma ;

namedArg : ID '=' arg -> ^('=' ID arg) ;

list: {input.LA(2)==RBRACK}? // hush warning; [] special case
  lb='[' ']' -> LIST[$lb]
 | lb='[' listElement ( ',' listElement )* ']' -> ^(LIST[$lb] listElement*)
 ;

listElement : exprNoComma | -> NULL ;
```

Found in path(s):
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/STParser.g
No license file was found, but licenses were detected in source scan.

```
/*
* [The "BSD license"]
*  Copyright (c) 2011 Terence Parr
*  All rights reserved.
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. The name of the author may not be used to endorse or promote products
*     derived from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
*  IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
*  OF MERCHANTABILITY
 AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
*  IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
*  INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
*  NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
```

* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/gui/STViewFrame.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/Coordinate.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/gui/JTreeASTModel.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/CompiledST.java
*
 /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/AggregateModelAdaptor.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/STErrorListener.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/AttributeRenderer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/MapModelAdaptor.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/STWriter.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/STGroup.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/MultiMap.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/ArrayIterator.java
*
 /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STMessage.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/Interval.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/GroupLexer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/STGroupDir.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/STGroupFile.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/STLexer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/debug/IndentEvent.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STCompiletimeMessage.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-

jar/org/stringtemplate/v4/STGroupString.java
*
/opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/ErrorBuffer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STModelAdaptor.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/debug/ConstructionEvent.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/gui/STViz.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/gui/JTreeSTModel.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/ErrorManager.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/ObjectModelAdaptor.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/debug/EvalTemplateEvent.java
*
/opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/CodeGenerator.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/gui/JTreeScopeStackModel.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/NoIndentWriter.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/DateRenderer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/StringTable.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/AutoIndentWriter.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/BytecodeDisassembler.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/GroupParser.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/CompilationState.java
*
/opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STGroupCompiletimeMessage.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/Aggregate.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/ModelAdaptor.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/Misc.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STNoSuchAttributeException.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/ST.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-

jar/org/stringtemplate/v4/misc/STNoSuchPropertyException.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/Compiler.java
*
 /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/InstanceScope.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STRuntimeMessage.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/FormalArgument.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/StringRenderer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/Bytecode.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/NumberRenderer.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/debug/EvalExprEvent.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/Interpreter.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/debug/AddAttributeEvent.java
*
 /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/STLexerMessage.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/misc/ErrorType.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/compiler/STException.java
* /opt/cola/permits/175540161_1695324887.5645561/0/st4-4-0-8-sources-jar/org/stringtemplate/v4/debug/InterpEvent.java

1.122 netty 3.10.5

1.122.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too,
but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give
the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to
the existence of
any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination
fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only,
so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC
LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included
without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately
publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the
modified Library refers to a function or a
table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works
in
themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you

distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this
License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must
be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the

Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding

machine-readable
source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form)
with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you

cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices.
Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General
Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by

the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the
free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL
OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

The Netty Project
=================

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This
 product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:
   * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of
zlib in pure Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.jzlib.txt (BSD Style License)
 * HOMEPAGE:
   * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based
WebSocket and HTTP
 server:

 * LICENSE:
   * license/LICENSE.webbit.txt (BSD License)
 * HOMEPAGE:
   * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

- LICENSE:
  - license/LICENSE.protobuf.txt (New BSD License)
- HOMEPAGE:
  - http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

- LICENSE:
  - license/LICENSE.bouncycastle.txt (MIT License)
- HOMEPAGE:
  - http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

- LICENSE:
  - license/LICENSE.slf4j.txt (MIT License)
- HOMEPAGE:
  - http://www.slf4j.org/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

- LICENSE:
  - license/LICENSE.commons-logging.txt (Apache License 2.0)
- HOMEPAGE:
  - http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

- LICENSE:
  - license/LICENSE.log4j.txt (Apache License 2.0)
- HOMEPAGE:
  - http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

- LICENSE:
  - license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
- HOMEPAGE:

  * http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

 * LICENSE:
   * license/LICENSE.felix.txt (Apache License 2.0)
 * HOMEPAGE:
   * http://felix.apache.org/

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate
 of all present
and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
                       (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.
The person or persons who have associated work with this document (the
"Dedicator" or "Certifier") hereby either (a) certifies that, to the best of
his knowledge, the work of authorship identified is in the public domain of
the country from which the work is published, or (b) hereby dedicates whatever
copyright the dedicators holds in the work of authorship identified below (the
"Work") to the public domain. A certifier, moreover, dedicates any copyright
interest he may have in the associated work, and for these purposes, is
described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this
work. Certifier recognizes that his good faith efforts may not shield him from
liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to
the detriment of the Dedicator's heirs and successors. Dedicator intends this
dedication to be an overt act of relinquishment in perpetuity
of all present
and future rights under copyright law, whether vested or contingent, in the
Work. Dedicator understands that such relinquishment of all rights includes
the relinquishment of all rights to enforce (by lawsuit or otherwise) those
copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be
freely reproduced, distributed, transmitted, used, modified, built upon, or
otherwise exploited by anyone for any purpose, commercial or non-commercial,
and in any way, including by methods that have not yet been invented or
conceived.
(BSD License: http://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or

without modification, are permitted provided that the
following conditions are met:

* Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other
materials provided with the distribution.

* Neither the name of the Webbit nor the names of
its contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
/*
* Copyright (c) 2004-2007 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT,
INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE
BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.123 abego-treelayout-core 1.0.1

1.123.1 Available under license :

[The "BSD license"]
Copyright (c) 2011, abego Software GmbH, Germany (http://www.abego.org)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the abego Software GmbH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

1.124 netty 4.1.11

1.124.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/benchmarkserver/WebSocketServerBenchmarkPage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlDecoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttConnectMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttMessageType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpHeaderNames.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollSocketChannel.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketServerExtensionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyProxiedProtocol.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspHeaderValues.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlNamespace.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkAlpnApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlDocumentEnd.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkSslContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollEventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyProtocolException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionUtil.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttCodecUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslSessionContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ClientEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ThreadDeathWatcher.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/server/HelloWorldHttp1Handler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlDocumentStart.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollChannelOption.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkSslServerContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/AsciiString.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlSpace.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/ssl/SslContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollTcpInfo.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyProtocolVersion.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslSessionStats.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsNameResolverContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttUnsubscribeMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/IpFilterRuleType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/AbstractEpollChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/DefaultLastStompContentSubframe.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/LzfDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpHeaderValues.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanStageEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateFrameClientExtensionHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/server/package-info.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttSubAckPayload.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslDefaultApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/ProxyHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SniHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/IntegerHolder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/ThreadLocalInsecureRandom.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/AbstractSocksMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/PromiseNotifier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketClientExtensionHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttUnsubAckMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameDecoderDelegate.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtension.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/DefaultStompHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsNameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlAttribute.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/HttpProxyHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/server/SpdyServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2Encoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ResourceLeakHint.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/InsecureTrustManagerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlComment.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttConnectPayload.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/WebSocketServerCompressionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/ssl/util/BouncyCastleSelfSignedCertGenerator.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/NameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/benchmarkserver/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2Decoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlElementStart.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerFrameDeflateEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialRequestDecoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Mapping.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/benchmarkserver/WebSocketServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketClientExtension.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/DefaultStompContentSubframe.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerFrameDeflateDecoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/ClientCookieEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/FastLz.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/AddressResolverGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttTopicSubscription.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyConstants.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspHeaderNames.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompConstants.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/benchmarkserver/WebSocketServerHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JettyAlpnSslEngine.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Lz4FrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandRequestDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollServerSocketChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsQueryContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/client/WebSocketClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/InternalThreadLocalMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyProtocolException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequestDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlEntityReference.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/CipherSuiteConverter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/ServerCookieEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/LzmaFrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkNpnApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/client/HttpResponseClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/LastStompContentSubframe.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/Socks4ProxyHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollDatagramChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5ClientEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/ProxyConnectionEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/UnixChannelOption.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/stomp/StompHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompCommand.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSsl.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/Epoll.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/SimpleTrustManagerFactory.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttConnAckMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/stomp/StompClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompHeadersSubframe.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/memcache/binary/MemcacheClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompSubframeDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/AbstractEventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SslProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockZlibDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/DefaultStompFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2Rand.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5ServerEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompSubframeEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/IpFilterRule.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/IdentityCipherSuiteFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslServerContext.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkSslClientContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttConnectReturnCode.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkDefaultApplicationProtocolNegotiator.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2SecurityUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/Java7SslParametersUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2DivSufSort.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttMessageFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateFrameServerExtensionHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JettyNpnSslEngine.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/PromiseAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelFactory.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/server/SpdyServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2Constants.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2MTFAndRLE2StageEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/server/SpdyOrHttpHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ServerEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslEngine.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/client/SpdyClientInitializer.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlElementEnd.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslX509Certificate.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttSubscribeMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkSslSession.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/ssl/SslUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/OpenJdkSelfSignedCertGenerator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttSubscribePayload.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/LzfEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Lz4FrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpMessageUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketServerExtension.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/benchmarkserver/WebSocketServerInitializer.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/client/SpdyClientStreamIdHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/client/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompSubframeAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/CipherSuiteFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlCdata.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlCharacters.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/NativeDatagramPacketArray.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/file/FileServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2MoveToFrontTable.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttPubAckMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SupportedCipherSuiteFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/Socks5ProxyHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4Message.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketServerExtensionHandshaker.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/PendingWriteQueue.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsAddressResolverGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateServerExtensionHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/WebSocketClientCompressionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlDTD.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/FastLzFrameEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/SimpleNameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttPublishVariableHeader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/DefaultStompHeadersSubframe.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttIdentifierRejectedException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/stomp/StompClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2BlockDecompressor.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketExtensionData.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslNpnApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/DomainNameMapping.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/AbstractSocks4Message.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttDecoder.java

*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/RuleBasedIpFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslClientContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/Utf8FrameValidator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlElement.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttUnacceptableProtocolVersionException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DecoderResultProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/proxy/ProxyConnectException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttPublishMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/json/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/server/Http2ServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Lz4Constants.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/AbstractRemoteAddressFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslEngineMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/NoopAddressResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttMessageIdVariableHeader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyMessageDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2BitWriter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/MessageAggregationException.java
*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/PemReader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/memcache/binary/MemcacheClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttConnAckVariableHeader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/AsciiHeadersEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/server/SpdyServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2InboundFrameLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/AbstractSocks5Message.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/UniqueIpFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/SnappyFramedEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkSslEngine.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/NoopAddressResolverGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/json/JsonObjectDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlProcessingInstruction.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ipfilter/IpSubnetFilterRule.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2OutboundFrameLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/client/SpdyFrameLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpHeadersEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsServerAddresses.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompSubframe.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/spdy/client/SpdyClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/example/http2/helloworld/server/Http2Server.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/WebSocketClientExtensionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpChunkedInput.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/FastLzFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/UnpaddedInternalThreadLocalMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2BlockCompressor.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/FastThreadLocal.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/IovArray.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/DeflateEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/haproxy/HAProxyCommand.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/SnappyFramedDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttUnsubscribePayload.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/CertificateVerifier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslApplicationProtocolNegotiator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttSubAckMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/embedded/EmbeddedChannelId.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttVersion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficCounter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2BitReader.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttFixedHeader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/FingerprintTrustManagerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5Message.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttConnectVariableHeader.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/stomp/StompContentSubframe.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/extensions/compression/PerMessageDeflateClientExtensionHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Crc32.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ReflectiveChannelFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/JdkBaseApplicationProtocolNegotiator.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Bzip2HuffmanStageDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpStatusClass.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/NativeLibraryLoader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslServerSessionContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockRawDecoder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/CleanerJava6.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnsafeDirectSwappedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/FastThreadLocalThread.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultChannelHandlerContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnsafeHeapSwappedByteBuf.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/META-INF/maven/io.netty/netty-all/pom.xml
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/sctp/SctpMessageToMessageDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtMessageAcceptorChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/sctp/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtByteRendezvousChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/UdtChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/ChannelTrafficShapingHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtByteConnectorChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtMessageConnectorChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/TrafficCounter.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtAcceptorChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/GlobalTrafficShapingHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/sctp/SctpMessageCompletionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/sctp/SctpInboundByteStreamHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/NioUdtByteAcceptorChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/udt/nio/NioUdtMessageRendezvousChannel.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/server/Http2OrHttpHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2LifecycleManager.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2HeadersEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameListener.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Headers.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2LocalFlowController.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2NoMoreStreamIdsException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/ByteCollections.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/ShortObjectHashMap.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/InboundHttp2ToHttpAdapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Error.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DefaultHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/LongObjectMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ClientUpgradeCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/IntObjectHashMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Connection.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Settings.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ServerUpgradeCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/LongObjectHashMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/ByteObjectHashMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/ObjectUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/ByteObjectMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/CharObjectMap.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/CharCollections.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FlowController.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/IntCollections.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpClientUpgradeHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameReader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DelegatingDecompressorFrameListener.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/client/Http2Client.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Flags.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionAdapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/IntObjectMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/util/collection/ShortObjectMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HttpToHttp2ConnectionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/EmptyHeaders.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2RemoteFlowController.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameSizePolicy.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2Connection.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpServerUpgradeHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameListenerDecorator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/server/HelloWorldHttp2Handler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/client/Http2ClientInitializer.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/CompressorHttp2ConnectionEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2FrameReader.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/EmptyHttp2Headers.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/client/Http2SettingsHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2EventAdapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameAdapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameWriter.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2HeadersDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/Http2ExampleUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2LocalFlowController.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2Headers.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2HeadersDecoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Stream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttQoS.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/LongCollections.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2DataWriter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2CodecUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2FrameWriter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2ConnectionDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameTypes.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/client/HttpResponseHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/CharObjectHashMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2HeadersEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2RemoteFlowController.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Exception.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/collection/ShortCollections.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/Headers.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2ConnectionEncoder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledHeapByteBuf.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisArrayAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/multiplex/server/Http2OrHttpHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/DefaultLastBulkStringRedisContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/IntegerRedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisCodecUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/AbstractHttp2StreamChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisConstants.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/FullBulkStringRedisMessage.java

*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/FixedRedisMessagePool.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/BulkStringHeaderRedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/BulkStringRedisContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/ArrayRedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/AbstractStringRedisMessage.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/redis/RedisClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisMessageType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/ArrayHeaderRedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/multiplex/server/HelloWorldHttp2Handler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/ErrorRedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisBulkStringAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisCodecException.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/DefaultBulkStringRedisContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/RedisMessagePool.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/SimpleStringRedisMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/redis/LastBulkStringRedisContent.java
No license file was found, but licenses were detected in source scan.

```
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* Handles an I/O event or intercepts an I/O operation, and forwards it to its next handler in
* its {@link ChannelPipeline}.
*
* <h3>Sub-types</h3>
* <p>
* {@link ChannelHandler} itself does not provide many methods, but you usually have to implement one of its
subtypes:
* <ul>
* <li>{@link
 ChannelInboundHandler} to handle inbound I/O events, and</li>
* <li>{@link ChannelOutboundHandler} to handle outbound I/O operations.</li>
* </ul>
* </p>
* <p>
* Alternatively, the following adapter classes are provided for your convenience:
* <ul>
* <li>{@link ChannelInboundHandlerAdapter} to handle inbound I/O events,</li>
* <li>{@link ChannelOutboundHandlerAdapter} to handle outbound I/O operations, and</li>
* <li>{@link ChannelDuplexHandler} to handle both inbound and outbound events</li>
* </ul>
* </p>
* <p>
* For more information, please refer to the documentation of each subtype.
* </p>
*
* <h3>The context object</h3>
* <p>
* A {@link ChannelHandler} is provided with a {@link ChannelHandlerContext}
* object.  A {@link ChannelHandler} is supposed to interact with the
* {@link ChannelPipeline} it belongs to via a context object.  Using the
* context object, the {@link ChannelHandler} can pass events upstream or
* downstream, modify the pipeline dynamically,
 or store the information
* (using {@link AttributeKey}s) which is specific to the handler.
*
* <h3>State management</h3>
*
```

```
* A {@link ChannelHandler} often needs to store some stateful information.
* The simplest and recommended approach is to use member variables:
* <pre>
* public interface Message {
*     // your methods here
* }
*
* public class DataServerHandler extends {@link SimpleChannelInboundHandler}&lt;Message&gt; {
*
*     <b>private boolean loggedIn;</b>
*
*     {@code @Override}
*     public void channelRead0({@link ChannelHandlerContext} ctx, Message message) {
*         {@link Channel} ch = e.getChannel();
*         if (message instanceof LoginMessage) {
*             authenticate((LoginMessage) message);
*             <b>loggedIn = true;</b>
*         } else (message instanceof GetDataMessage) {
*             if (<b>loggedIn</b>) {
*                 ch.write(fetchSecret((GetDataMessage) message));
*             } else {
*
  fail();
*             }
*         }
*     }
*     ...
* }
* </pre>
* Because the handler instance has a state variable which is dedicated to
* one connection, you have to create a new handler instance for each new
* channel to avoid a race condition where a unauthenticated client can get
* the confidential information:
* <pre>
* // Create a new handler instance per channel.
* // See {@link ChannelInitializer#initChannel(Channel)}.
* public class DataServerInitializer extends {@link ChannelInitializer}&lt;{@link Channel}&gt; {
*     {@code @Override}
*     public void initChannel({@link Channel} channel) {
*         channel.pipeline().addLast("handler", <b>new DataServerHandler()</b>);
*     }
* }
*
* </pre>
*
* <h4>Using {@link AttributeKey}s</h4>
*
* Although it's recommended to use member variables to store the state of a
* handler, for some reason you might not want to create many handler instances.
* In such a case, you can use {@link AttributeKey}s which
 is provided by
* {@link ChannelHandlerContext}:
* <pre>
* public interface Message {
*     // your methods here
* }
*
* {@code @Sharable}
* public class DataServerHandler extends {@link SimpleChannelInboundHandler}&lt;Message&gt; {
*     private final {@link AttributeKey}&lt;{@link Boolean}&gt; auth =
*           {@link AttributeKey#valueOf(String) AttributeKey.valueOf("auth")};
*
*     {@code @Override}
*     public void channelRead({@link ChannelHandlerContext} ctx, Message message) {
*         {@link Attribute}&lt;{@link Boolean}&gt; attr = ctx.attr(auth);
*         {@link Channel} ch = ctx.channel();
*         if (message instanceof LoginMessage) {
*             authenticate((LoginMessage) o);
*             <b>attr.set(true)</b>;
*         } else (message instanceof GetDataMessage) {
*             if (<b>Boolean.TRUE.equals(attr.get())</b>) {
*                 ch.write(fetchSecret((GetDataMessage) o));
*             } else {
*                 fail();
*
          }
*         }
*     }
*     ...
* }
* </pre>
* Now that the state of the handler is attached to the {@link ChannelHandlerContext}, you can add the
* same handler instance to different pipelines:
* <pre>
* public class DataServerInitializer extends {@link ChannelInitializer}&lt;{@link Channel}&gt; {
*
*     private static final DataServerHandler <b>SHARED</b> = new DataServerHandler();
*
*     {@code @Override}
*     public void initChannel({@link Channel} channel) {
*         channel.pipeline().addLast("handler", <b>SHARED</b>);
*     }
* }
* </pre>
*
*
```

```
* <h4>The {@code @Sharable} annotation</h4>
* <p>
* In the example above which used an {@link AttributeKey},
* you might have noticed the {@code @Sharable} annotation.
* <p>
* If a {@link ChannelHandler} is annotated with the {@code @Sharable}
* annotation, it means you can create an instance of the handler just once and
* add it to one or more {@link ChannelPipeline}s multiple times without
* a race
 condition.
* <p>
* If this annotation is not specified, you have to create a new handler
* instance every time you add it to a pipeline because it has unshared state
* such as member variables.
* <p>
* This annotation is provided for documentation purpose, just like
* <a href="http://www.javaconcurrencyinpractice.com/annotations/doc/">the JCIP annotations</a>.
*
* <h3>Additional resources worth reading</h3>
* <p>
* Please refer to the {@link ChannelHandler}, and
* {@link ChannelPipeline} to find out more about inbound and outbound operations,
* what fundamental differences they have, how they flow in a  pipeline,  and how to handle
* the operation in your application.
*/
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelHandler.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/*
*  Licensed to the Apache Software Foundation (ASF) under one or more
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/SSLContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/SSL.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/Buffer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/Library.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/SctpMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/SctpChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/sctp/SctpServerChannelConfig.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/nio/NioSctpChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/AbstractTrafficShapingHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockProtocol.proto
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/nio/NioSctpServerChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/SctpNotificationHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/SctpChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/DefaultSctpChannelConfig.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/SctpServerChannel.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsPtrRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslCertificateException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DatagramPacketEncoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2HeadersFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/util/internal/ConstantTimeUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/ThrowableUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2GoAwayFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/CertificateRequestedCallback.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/UnstableApi.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ServerDowngrader.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ResourceLeakDetectorFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/UnaryPromiseNotifier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/RejectedExecutionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueServerDomainSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2StreamClosedEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueDomainSocketChannelConfig.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueChannelOption.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2MultiplexCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/CodecOutputList.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/ErrorsStaticallyReferencedJniMethods.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsOptPseudoRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/flush/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/AbstractKQueueStreamChannel.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2StreamActiveEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/Java8SslUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/CompressionUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueSocketChannelConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2ResetFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DefaultDnsCache.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/VoidChannelGroupFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/Limits.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2WindowUpdateFrame.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueEventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2StreamChannelBootstrap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueServerSocketChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2HeadersFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/DefaultSmtpRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/Log4J2Logger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/BooleanSupplier.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueDatagramChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/LastSmtpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/NonStickyEventExecutorGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/buffer/PooledSlicedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/Log4J2LoggerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/PemPrivateKey.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/NoopDnsCache.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutorChooserFactory.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslKeyMaterialManager.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/InflightNameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/WriteBufferWaterMark.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/flow/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/NativeLibraryUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/ByteBufChecksum.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultSelectStrategy.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/OutOfDirectMemoryError.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpRequests.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2StreamStateEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueRecvByteAllocatorHandle.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsOptEcsRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ResourceLeakTracker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/MacAddressUtil.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledDuplicatedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/WrappedUnpooledUnsafeDirectByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/ReadOnlyHttp2Headers.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/PemValue.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/RejectedExecutionHandlers.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledUnsafeNoCleanerDirectByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/WrappedCompositeByteBuf.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelOutboundInvoker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/PemX509Certificate.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/FileUploadUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/LimitsStaticallyReferencedJniMethods.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpServerKeepAliveHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/PeerCredentials.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/multiplex/server/Http2ServerInitializer.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultSelectStrategyFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/Native.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/redis/RedisClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractUnpooledSlicedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelInboundInvoker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2DataFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/SocketWritableByteChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2DataFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/string/LineEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ResetFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/bootstrap/BootstrapConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/NativeLongArray.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/EventExecutorChooserFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/RoundRobinInetAddressResolver.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AdvancedLeakAwareCompositeByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/SimpleLeakAwareCompositeByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Codec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpRequestEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/string/LineSeparator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/UnorderedThreadPoolEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueServerChannelConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DatagramPacketDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/AbstractKQueueChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/AbstractHttp2StreamFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/AbstractDnsOptPseudoRrRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueEventArray.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2WindowUpdateFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/RoundRobinDnsAddressResolverGroup.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DateFormatter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/InboundHttpToHttp2Adapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/OrderedEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/NativeStaticallyReferencedJniMethods.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslClientContext.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/flush/FlushConsolidationHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/DomainNameMappingBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/PromiseCombiner.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/DelegatingSslContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslExtendedKeyMaterialManager.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractPooledDerivedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/X509TrustManagerWrapper.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2StreamFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/AbstractBootstrapConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueDatagramChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/BsdSocket.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DuplexChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsPtrRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/AbstractHttp2StreamStateEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueDomainSocketChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/SmtpCommand.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsCache.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/SocketUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketChunkedInput.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsOptEcsRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/SelectStrategy.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/LinuxSocket.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueue.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/IntSupplier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueServerSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/SelectStrategyFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/PemEncoded.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/PromiseNotificationUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2FrameCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/DefaultLastSmtpContent.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2Frame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/AcceptFilter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DefaultHttp2GoAwayFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslEngine.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ReferenceCountedOpenSslServerContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/SessionTicketKey.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/multiplex/server/Http2Server.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/DefaultSmtpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/ServerBootstrapConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/AbstractKQueueServerChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/kqueue/KQueueStaticallyReferencedJniMethods.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/smtp/DefaultSmtpResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/PreferHeapByteBufAllocator.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolSubpageMetric.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufVarint32LengthFieldPrepender.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/TcpMd5Util.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsRecordEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/CookieUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/DomainSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ClientAuth.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/RotationalDnsServerAddresses.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/Launcher.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/DefaultPriorityQueue.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsNameResolverException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollServerDomainSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/resolver/InetSocketAddressResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnsafeByteBufUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollRecvByteAllocatorHandle.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/AbstractEpollServerChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/ServerCookieEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsSection.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/DefaultCookie.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/PriorityQueueNode.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/ClientCookieEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DatagramDnsQueryEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/ChannelHealthChecker.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/helloworld/server/HelloWorldHttp2HandlerBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DatagramDnsResponseEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/AbstractDnsMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/LongCounter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/SocksPortUnificationServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollDomainSocketChannelConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ProtocolDetectionState.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultMaxBytesRecvByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/ShuffledDnsServerAddressStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslSessionTicketKey.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/EmptyHttpHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/Errors.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DefaultDnsServerAddresses.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/CompositeNameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/FileDescriptor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsRawRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/DomainMappingBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollServerChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/MaxMessagesRecvByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/ChannelPool.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/CombinedHttpHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/CookieHeaderNames.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/ChannelPoolHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractUnsafeSwappedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolChunkListMetric.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufVarint32FrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsOpCode.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufDecoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsRecordDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultMaxMessagesRecvByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/HashingStrategy.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/unix/ServerDomainSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsRawRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollRecvByteAllocatorStreamingHandle.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpExpectationFailedEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/MaxBytesRecvByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/FixedChannelPool.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/DefaultNameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/Http2Server.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/HttpServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DatagramDnsQuery.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/HeapByteBufUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ThreadProperties.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/UnixChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DatagramDnsQueryDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/Http2RequestHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/UnsupportedValueConverter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsMessageUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/ImageCache.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsNameResolverBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/AbstractEpollStreamChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/SimpleChannelPool.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsQueryContextManager.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/HeadersUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsQuestion.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/AbstractChannelPoolHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/ChannelPoolMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/CookieEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolAccessor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/DefaultHostsFileEntriesResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/SingletonDnsServerAddresses.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/sctp/multihoming/SctpMultiHomingEchoClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/AsyncMapping.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollMode.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolArenaMetric.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsRecordEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/DatagramSocketAddress.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/CharSequenceMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/sctp/multihoming/SctpMultiHomingEchoServer.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HttpToHttp2ConnectionHandlerBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/HostsFileParser.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AddressDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/ServerCookieDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/Http2OrHttpHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/dns/DefaultDnsQuestion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/Html.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsRecordType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/DomainSocketAddress.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/InetNameResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledSlicedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/NativeInetAddress.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/DomainSocketReadMode.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/AbstractAddressResolver.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/AbstractDnsRecord.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledUnsafeHeapByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/AbstractScheduledEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufEncoderNano.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/AbstractHttp2ConnectionHandlerBuilder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/Cookie.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/DefaultAddressResolverGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollEventArray.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/Http1RequestHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNegotiationHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http2/tiles/FallbackRequestHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/DomainSocketChannelConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/AddressResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ProtocolDetectionResult.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsRecordDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DatagramDnsResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DefaultDnsQuery.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolChunkMetric.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollDomainSocketChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ApplicationProtocolNames.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SslContextBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsCacheEntry.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledDuplicatedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpScheme.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledUnsafeHeapByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsServerAddressStream.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/ClientCookieDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/unix/Socket.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/HostsFileEntriesResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AddressEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/http/cookie/CookieUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionHandlerBuilder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OpenSslJavaxX509Certificate.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DatagramDnsResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsQuery.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cookie/CookieDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/dns/DnsResponseCode.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/ProtobufDecoderNano.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/SequentialDnsServerAddressStream.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/pool/AbstractChannelPoolMap.java
No license file was found, but licenses were detected in source scan.

// Try the OpenJDK's proprietary implementation.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/SelfSignedCertificate.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/ThreadLocalRandom.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DecoratingHttp2FrameWriter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2PromisedRequestVerifier.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/WeightedFairQueueByteDistributor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/http2/Http2StreamVisitor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/InboundHttp2ToHttpAdapterBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DefaultHeadersImpl.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DecoratingHttp2ConnectionEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/StreamByteDistributor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/UniformStreamByteDistributor.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ValueConverter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/DecoratingHttp2ConnectionDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ByteProcessor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/StreamBufferingEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/AbstractInboundHttp2ToHttpAdapterBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/CoalescingBufferQueue.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/MathUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/CharSequenceValueConverter.java
No license file was found, but licenses were detected in source scan.

*/
/**
* Utility class for {@link ByteBuf} that encodes and decodes to and from
* <a href="http://en.wikipedia.org/wiki/Base64">Base64</a> notation.
* <p>
*
 The encoding and decoding algorithm in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/base64/Base64.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version 2.0 (the
* "License"); you may not use this file except in compliance with the License. You may obtain a
* copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
// https://tools.ietf.org/html/rfc7540#section-8.1.2.2 makes a special exception for TE

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HttpConversionUtil.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2016 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

```
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/flow/FlowControlHandler.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
* A decoder that splits the received {@link ByteBuf}s dynamically by the
* value of the length field in the message.  It is particularly useful when you
* decode a binary message which has an integer header field that represents the
* length of the message body or the whole message.
*
 <p>
* {@link LengthFieldBasedFrameDecoder} has many configuration parameters so
* that it can decode any message with a length field, which is often seen in
* proprietary client-server protocols. Here are some example that will give
* you the basic idea on which option does what.
*
* <h3>2 bytes length field at offset 0, do not strip header</h3>
*
* The value of the length field in this example is <tt>12 (0x0C)</tt> which
* represents the length of "HELLO, WORLD".  By default, the decoder assumes
* that the length field represents the number of the bytes that follows the
* length field.  Therefore, it can be decoded with the simplistic parameter
* combination.
* <pre>
* <b>lengthFieldOffset</b>   = <b>0</b>
* <b>lengthFieldLength</b>   = <b>2</b>
* lengthAdjustment    = 0
* initialBytesToStrip = 0 (= do not strip header)
*
* BEFORE DECODE (14 bytes)         AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content
 |----->| Length | Actual Content |
* | 0x000C | "HELLO, WORLD" |      | 0x000C | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
*
* <h3>2 bytes length field at offset 0, strip header</h3>
*
* Because we can get the length of the content by calling
* {@link ByteBuf#readableBytes()}, you might want to strip the length
* field by specifying <tt>initialBytesToStrip</tt>.  In this example, we
* specified <tt>2</tt>, that is same with the length of the length field, to
* strip the first two bytes.
* <pre>
* lengthFieldOffset   = 0
* lengthFieldLength   = 2
* lengthAdjustment    = 0
* <b>initialBytesToStrip</b> = <b>2</b> (= the length of the Length field)
*
* BEFORE DECODE (14 bytes)         AFTER DECODE (12 bytes)
* +--------+----------------+      +----------------+
* | Length | Actual Content |----->| Actual Content |
* | 0x000C | "HELLO, WORLD" |      | "HELLO, WORLD" |
* +--------+----------------+      +----------------+
 * </pre>
*
* <h3>2 bytes length field at offset 0, do not strip header, the length field
*     represents the length of the whole message</h3>
*
* In most cases, the length field represents the length of the message body
* only, as shown in the previous examples.  However, in some protocols, the
* length field represents the length of the whole message, including the
* message header.  In such a case, we specify a non-zero
* <tt>lengthAdjustment</tt>.  Because the length value in this example message
* is always greater than the body length by <tt>2</tt>, we specify <tt>-2</tt>
* as <tt>lengthAdjustment</tt> for compensation.
* <pre>
* lengthFieldOffset   =  0
* lengthFieldLength   =  2
* <b>lengthAdjustment</b>    = <b>-2</b> (= the length of the Length field)
* initialBytesToStrip =  0
*
* BEFORE DECODE (14 bytes)         AFTER DECODE (14 bytes)
* +--------+----------------+      +--------+----------------+
* | Length | Actual Content |----->| Length | Actual
 Content |
* | 0x000E | "HELLO, WORLD" |      | 0x000E | "HELLO, WORLD" |
* +--------+----------------+      +--------+----------------+
* </pre>
*
* <h3>3 bytes length field at the end of 5 bytes header, do not strip header</h3>
*
* The following message is a simple variation of the first example.  An extra
* header value is prepended to the message.  <tt>lengthAdjustment</tt> is zero
* again because the decoder always takes the length of the prepended data into
* account during frame length calculation.
* <pre>
* <b>lengthFieldOffset</b>   = <b>2</b> (= the length of Header 1)
* <b>lengthFieldLength</b>   = <b>3</b>
* lengthAdjustment    = 0
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes)                      AFTER DECODE (17 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* | Header 1 |  Length  | Actual Content |----->| Header 1 |  Length  | Actual Content |
* |  0xCAFE  | 0x00000C | "HELLO, WORLD" |      |
  0xCAFE  | 0x00000C | "HELLO, WORLD" |
* +----------+----------+----------------+      +----------+----------+----------------+
* </pre>
*
* <h3>3 bytes length field at the beginning of 5 bytes header, do not strip header</h3>
*
* This is an advanced example that shows the case where there is an extra
* header between the length field and the message body.  You have to specify a
* positive <tt>lengthAdjustment</tt> so that the decoder counts the extra
* header into the frame length calculation.
* <pre>
* lengthFieldOffset   = 0
* lengthFieldLength   = 3
* <b>lengthAdjustment</b>    = <b>2</b> (= the length of Header 1)
* initialBytesToStrip = 0
*
* BEFORE DECODE (17 bytes)                      AFTER DECODE (17 bytes)
* +----------+----------+----------------+      +----------+----------+----------------+
* |  Length  | Header 1 | Actual Content |----->|  Length  | Header 1 | Actual Content |
* | 0x00000C |  0xCAFE  | "HELLO, WORLD" |      | 0x00000C |  0xCAFE  | "HELLO,
 WORLD" |
* +----------+----------+----------------+      +----------+----------+----------------+
* </pre>
*
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
*     strip the first header field and the length field</h3>
*
* This is a combination of all the examples above.  There are the prepended
* header before the length field and the extra header after the length field.
* The prepended header affects the <tt>lengthFieldOffset</tt> and the extra
* header affects the <tt>lengthAdjustment</tt>.  We also specified a non-zero
* <tt>initialBytesToStrip</tt> to strip the length field and the prepended
* header from the frame.  If you don't want to strip the prepended header, you
* could specify <tt>0</tt> for <tt>initialBytesToSkip</tt>.
* <pre>
* lengthFieldOffset   = 1 (= the length of HDR1)
* lengthFieldLength   = 2
* <b>lengthAdjustment</b>    = <b>1</b> (= the length of HDR2)
* <b>initialBytesToStrip</b> = <b>3</b> (= the length of HDR1 + LEN)
 *
* BEFORE DECODE (16 bytes)                       AFTER DECODE (13 bytes)
* +------+--------+------+----------------+      +------+----------------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x000C | 0xFE | "HELLO, WORLD" |      | 0xFE | "HELLO, WORLD" |
* +------+--------+------+----------------+      +------+----------------+
* </pre>
*
* <h3>2 bytes length field at offset 1 in the middle of 4 bytes header,
*     strip the first header field and the length field, the length field
*     represents the length of the whole message</h3>
*
* Let's give another twist to the previous example.  The only difference from
* the previous example is that the length field represents the length of the
* whole message instead of the message body, just like the third example.
* We have to count the length of HDR1 and Length into <tt>lengthAdjustment</tt>.
* Please note that we don't need to take the length of HDR2 into account
* because
 the length field already includes the whole header length.
* <pre>
* lengthFieldOffset   =  1
* lengthFieldLength   =  2
* <b>lengthAdjustment</b>    = <b>-3</b> (= the length of HDR1 + LEN, negative)
* <b>initialBytesToStrip</b> = <b> 3</b>
*
* BEFORE DECODE (16 bytes)                       AFTER DECODE (13 bytes)
* +------+--------+------+----------------+      +------+----------------+
* | HDR1 | Length | HDR2 | Actual Content |----->| HDR2 | Actual Content |
* | 0xCA | 0x0010 | 0xFE | "HELLO, WORLD" |      | 0xFE | "HELLO, WORLD" |
* +------+--------+------+----------------+      +------+----------------+
* </pre>
* @see LengthFieldPrepender
*/
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/LengthFieldBasedFrameDecoder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket13FrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket08FrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket07FrameEncoder.java
No license file was found, but licenses were detected in source scan.

*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/traffic/GlobalChannelTrafficShapingHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2013 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelProgressiveFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyStreamStatus.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/Future.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ImmediateExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelProgressivePromise.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/util/internal/EmptyArrays.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdySettingsFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/UdtChannelOption.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketProtocolHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/ConcurrentSet.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheClientCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/helloworld/HttpHelloWorldServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ResourceLeakException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/FullBinaryMemcacheRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/MemcacheContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/FullMemcacheMessage.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AddressType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/NoOpTypeParameterMatcher.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnreleasableByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/AbstractEventExecutorGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ConstantPool.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/ComposedLastHttpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksAuthStatus.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheObjectAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SslHandshakeCompletionEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ConnectTimeoutException.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/AbstractMemcacheObject.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/DefaultOioSocketChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyVersion.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockProtocol.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksMessageType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractDerivedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/SocksVersion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksProtocolVersion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/FixedCompositeByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ReferenceCounted.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheResponse.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyDataFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/EmptyByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheResponseEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ImmediateEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultPromise.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/AbstractBinaryMemcacheMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolHandshakeHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ResourceLeak.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ReadOnlyUnsafeDirectByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdySynReplyFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/spdy/SpdyHeaderBlockJZlibEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyWindowUpdateFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ProgressiveFuture.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/FullHttpResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cors/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultBinaryMemcacheResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheServerCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/OioSocketChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/ChannelMatcher.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdySynReplyFrame.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyRstStreamFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Recycler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultBinaryMemcacheRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/GenericProgressiveFutureListener.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdySessionHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AdvancedLeakAwareByteBuf.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdySynStreamFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/AbstractEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultProgressivePromise.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultFutureListeners.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/ChannelFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultChannelId.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/group/ChannelGroupException.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdySessionStatus.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/AbstractFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/SimpleLeakAwareByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelHandlerAdapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/rxtx/RxtxClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultFullBinaryMemcacheRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/MessageSizeEstimator.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/ChannelGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/DefaultUdtServerChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/PromiseTask.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientProtocolHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/ReadOnlyIterator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheOpcodes.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/FutureListener.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/FullHttpMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/GenericFutureListener.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandStatus.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ReferenceCountUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyPingFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyWindowUpdateFrame.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledUnsafeDirectByteBuf.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/AddressedEnvelope.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeadersFrame.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheResponseStatus.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyDataFrame.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/AbstractBinaryMemcacheDecoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpCodec.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/MemcacheObject.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/AbstractReferenceCounted.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultThreadFactory.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketFrameAggregator.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthStatus.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockZlibEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/AppendableCharSequence.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyStreamFrame.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/Native.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaders.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyRstStreamFrame.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksAuthScheme.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ReadOnlyByteBufferBuf.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheRequestEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksSubnegotiationVersion.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/ResourceLeakDetector.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockRawEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/FullHttpRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/RecyclableArrayList.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/CompleteFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/FullBinaryMemcacheResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksRequestType.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCmdType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketFrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ProgressivePromise.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelOutboundBuffer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/AbstractMemcacheObjectDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCmdStatus.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/SimpleChannelInboundHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/TypeParameterMatcher.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ScheduledFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdySynStreamFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultMessageSizeEstimator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyHeadersFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/package-info.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/DefaultChannelGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/spdy/SpdyGoAwayFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelId.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/AbstractEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufHolder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/PendingWrite.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksAddressType.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/rxtx/RxtxClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Version.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultFullHttpRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5AuthMethod.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/BinaryMemcacheRequestDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ScheduledFutureTask.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyCodecUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyStreamFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultFullHttpResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/DefaultByteBufHolder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/DefaultOioServerSocketChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/DefaultFullBinaryMemcacheResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/Promise.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/DefaultMemcacheContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/PlatformDependent0.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/spdy/DefaultSpdyGoAwayFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/binary/AbstractBinaryMemcacheEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/UdtServerChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/XmlFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultChannelProgressivePromise.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/WrappedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultAddressedEnvelope.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/AbstractMemcacheObjectEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdyPingFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractReferenceCountedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/OioServerSocketChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/package-info.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySet.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelProgressiveFutureListener.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufProcessor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/helloworld/HttpHelloWorldServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/JdkZlibDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/AbstractMemcacheObjectAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/ChannelMatchers.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHeaderBlockEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/ThreadPerTaskExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/DefaultLastMemcacheContent.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/LastMemcacheContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksResponseType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/memcache/MemcacheMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Crc32c.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License, version
* 2.0 (the "License"); you may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cors/CorsConfigBuilder.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2017 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/Cleaner.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/CleanerJava9.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/cors/OkResponseHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cors/CorsHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/cors/CorsConfig.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/rxtx/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/SctpChannelOption.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/rxtx/RxtxChannelConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/rxtx/RxtxChannelOption.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/rxtx/RxtxChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/rxtx/RxtxDeviceAddress.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/rxtx/DefaultRxtxChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/oio/OioByteStreamChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/UdtServerChannel.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/NoopDnsQueryLifecycleObserverFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownReadComplete.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufAllocatorMetricProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/MultiDnsServerAddressStreamProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/ssl/SslCloseCompletionEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsQueryLifecycleObserverFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/CleartextHttp2ServerUpgradeHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketScheme.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/mqtt/MqttMessageBuilders.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/ReflectionUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/NoopDnsQueryLifecycleObserver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsServerAddressStreamProviders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/HostsFileEntries.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/UncheckedBooleanSupplier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2CodecBuilder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ocsp/OcspClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SslCompletionEvent.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/SuppressForbidden.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/uptime/UptimeServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/SelectedSelectionKeySetSelector.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/ResolvedAddressTypes.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/UnixResolverDnsServerAddressStreamProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/LongAdderCounter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufAllocatorMetric.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/uptime/UptimeServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpServerExpectContinueHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/EmptyPriorityQueue.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DefaultDnsServerAddressStreamProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ocsp/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/SingletonDnsServerAddressStreamProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/AbstractSniHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsQueryLifecycleObserver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/NativeStaticallyReferencedJniMethods.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/resolver/dns/DnsServerAddressStreamProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/OptionalSslHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/ConscryptAlpnSslEngine.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/internal/tcnative/SniHostNameMatcher.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledByteBufAllocatorMetric.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/PriorityQueue.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/NettyRuntime.java
No license file was found, but licenses were detected in source scan.

*/
/**
* Enumeration of supported Base64 dialects.
* <p>
* The internal lookup tables in this class has been derived from
* <a href="http://iharder.sourceforge.net/current/java/base64/">Robert
 Harder's Public Domain
* Base64 Encoder/Decoder</a>.
*/

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/base64/Base64Dialect.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpObject.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/objectecho/ObjectEchoServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolChunkList.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/JdkZlibEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/echo/EchoServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpObjectAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ServerChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/example/discard/DiscardClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/Snappy.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/base64/Base64Decoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/proxy/HexDumpProxyInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Signal.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/base64/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/TimeoutException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvousBytes/ByteEchoPeerTwo.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/AbstractNioChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/DefaultChannelGroupFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/ZlibEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/ChannelBufferByteInput.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/snoop/HttpSnoopServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/AbstractChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/server/WebSocketIndexPageHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DelimiterBasedFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockClientHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/LimitingByteInput.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksInitRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/AbstractChannelHandlerContext.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/SystemPropertyUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/bytes/ByteEchoServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/example/http/websocketx/server/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/cors/HttpCorsServer.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/NioEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/IdleStateEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvous/MsgEchoPeerBase.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspVersions.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/securechat/SecureChatServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshakerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostRequestDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/VoidChannelPromise.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/base64/Base64Encoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ClientDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/SocksMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/server/WebSocketServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelDuplexHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/BlockingOperationException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/SslHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelInboundHandlerAdapter.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/LastHttpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpClientCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/echo/EchoClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/securechat/SecureChatClientInitializer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultHttpRequest.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutorGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksAuthResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/securechat/SecureChatClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/LocalChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/AbstractInternalLogger.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/SucceededChannelFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DatagramPacket.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/bytes/ByteEchoClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/file/HttpStaticFileServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket00FrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/sctp/SctpOutboundByteStreamHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/StringUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/AbstractNioByteChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandType.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/DuplicatedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/nio/NioSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/SocksServerConnectHandler.java
*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/discard/DiscardServerHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvous/MsgEchoPeerOne.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/ThreadLocalMarshallerProvider.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/Slf4JLoggerFactory.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/SocksServerUtils.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/InterfaceHttpPostRequestDecoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/UdtMessage.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ReadOnlyByteBuf.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/DefaultAttributeMap.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/DirectClientHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/snoop/HttpSnoopClient.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/upload/HttpUploadClientHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelPipelineException.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/NetUtil.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Timer.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/SocketChannel.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ThreadPerChannelEventLoop.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelPipeline.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/discard/DiscardServer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/DiskFileUpload.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/nio/ProtocolFamilyConverter.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelInboundHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/Attribute.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/FactorialServerInitializer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksResponse.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/localecho/LocalEchoClientHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspHeaders.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/util/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/SingleThreadEventLoop.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/qotm/QuoteOfTheMomentServerHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/logging/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultHttpContent.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/AbstractConstant.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelInitializer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/FixedLengthFrameDecoder.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker07.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5CommandResponse.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/CompositeByteBuf.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/message/MsgEchoClientHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ClassResolvers.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/oio/OioSctpServerChannel.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker13.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/oio/OioEventLoopGroup.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/WriteTimeoutException.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpResponse.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpContentEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpObjectEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpResponseDecoder.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/MixedFileUpload.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelPromise.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/bytes/ByteEchoServerHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledByteBufAllocator.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ClassResolver.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/package-info.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/MultithreadEventLoopGroup.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/TimerTask.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/nio/NioDatagramChannelConfig.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/proxy/HexDumpProxyBackendHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/DefaultEventExecutor.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/IllegalReferenceCountException.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/IdleStateHandler.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdySession.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksAuthRequestDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ClassLoaderClassResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DatagramChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/MessageAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/embedded/EmbeddedEventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/snoop/HttpSnoopClientInitializer.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/CompactObjectInputStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpData.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksInitResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/DecompressionException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/TextWebSocketFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledByteBufAllocator.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ThreadPerChannelEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5PasswordAuthResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultChannelPipeline.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/server/WebSocketFrameHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/protobuf/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/message/MsgEchoServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelFutureListener.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/CaseIgnoringComparator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/http/websocketx/BinaryWebSocketFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/CachingClassResolver.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvous/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksAuthRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/ChunkedWriteHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/portunification/PortUnificationServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/bytes/ByteArrayDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerProtocolHandshakeHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/LocalChannelRegistry.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/QueryStringDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpConstants.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/file/HttpStaticFileServer.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpContentDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketVersion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker00.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/DiskAttribute.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/SlicedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksMessage.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/snoop/HttpSnoopServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/WriteTimeoutHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockClientInitializer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/Log4JLoggerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/UnmarshallerProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/message/MsgEchoServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/SocketChannelConfig.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/ChunkedFile.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/localecho/LocalEcho.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/PongWebSocketFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/logging/LogLevel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/NioTask.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/ReadTimeoutHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/sctp/SctpEchoClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/nio/NioDatagramChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/SoftReferenceMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/SocksServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/CorruptedFrameException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/MemoryFileUpload.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelPromiseAggregator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpVersion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/example/http/helloworld/HttpHelloWorldServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/Bootstrap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/ChunkedStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/AbstractMemoryHttpData.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/CompleteChannelFuture.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/objectecho/ObjectEchoClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/SucceededFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/MarshallerProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/UnknownSocksRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpContentCompressor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/Delimiters.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultCookie.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/SwappedByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/DefaultSpdySettingsFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelFlushPromiseNotifier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/nio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/qotm/QuoteOfTheMomentServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultEventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/uptime/UptimeClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ByteToMessageCodec.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelMetadata.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/bytes/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/AbstractNioMessageChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5CommandRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketFrame.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/objectecho/ObjectEchoClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/CompressionException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandStatus.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/LocalServerChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCmdResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/FailedFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/ServerBootstrap.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/JdkLoggerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/FactorialClientInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultHttpMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/ZlibWrapper.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/Slf4JLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/cors/HttpCorsServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/oio/OioSctpChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/server/WebSocketServerIndexPage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PooledDirectByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ObjectDecoderInputStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/RecvByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/LocalAddress.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/qotm/QuoteOfTheMomentClient.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/ChunkedNioStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCommonUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/embedded/EmbeddedChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/localecho/LocalEchoServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/local/LocalEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultHttpObject.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/upload/HttpUploadServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocket00FrameDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ByteToMessageDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultLastHttpContent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DefaultDatagramChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/OioSocketChannel.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/LineBasedFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ObjectEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketHandshakeException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/DefaultSocks4CommandResponse.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/sctp/SctpEchoServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/sctp/SctpEchoServerHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshakerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/MessageToMessageCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspResponseStatuses.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollDatagramChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpMessage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandResponse.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/upload/HttpUploadClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/RelayHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/UnknownSocksResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5InitialRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/embedded/EmbeddedSocketAddress.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/telnet/TelnetClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/upload/HttpUploadClientInitializer.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Constant.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpResponseStatus.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/FailedChannelFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/CodecException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultFileRegion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/echo/EchoServer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvousBytes/ByteEchoPeerHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/EventLoopException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/nio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/InternalLogLevel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvous/MsgEchoPeerTwo.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvousBytes/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/JZlibEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/DefaultMarshallerProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/oio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpServerCodec.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/ContinuationWebSocketFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolThreadCache.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksInitResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/telnet/TelnetClientInitializer.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4CommandRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksMessageEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/MessageToMessageEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ObjectEncoderOutputStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelOutboundHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/handler/codec/socks/SocksAuthResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCmdRequestDecoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/ZlibUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/LengthFieldPrepender.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/Unpooled.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/MarshallingDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/IdleState.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/AbstractServerChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/EncoderException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/DefaultSocks4CommandRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/PrematureChannelClosureException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/Channel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolSubpage.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/bytes/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpRequestEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/Cookie.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/FileRegion.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/EventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/oio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/InterfaceHttpData.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DefaultServerSocketChannelConfig.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/CompatibleMarshallingEncoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/ChannelBufferByteOutput.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/UnsupportedMessageTypeException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/proxy/HexDumpProxyFrontendHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/securechat/SecureChatServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ReferenceMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvousBytes/ByteEchoPeerOne.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/EventExecutorGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/portunification/PortUnificationServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/file/HttpStaticFileServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/CloseWebSocketFrame.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/OioServerSocketChannel.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostMultipartRequestDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspObjectDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelPromiseNotifier.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/FactorialClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/ServerSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/UdtChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/ChannelGroupFutureListener.java
*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolChunk.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpHeaders.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/BigIntegerDecoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/NumberEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/CharsetUtil.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/xml/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/DefaultUnmarshallerProvider.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/string/StringEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/CompactObjectOutputStream.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/MultithreadEventExecutorGroup.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelOption.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/SocksServerInitializer.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/AttributeKey.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultChannelPromise.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/PingWebSocketFrame.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspResponseDecoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpResponseEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/oio/AbstractOioByteChannel.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/package-info.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufOutputStream.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/FixedRecvByteBufAllocator.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/MixedAttribute.java

*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/nio/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/AttributeMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/InternalAttribute.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBufInputStream.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/MarshallingEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/discard/DiscardClientHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCmdRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/telnet/TelnetClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/securechat/SecureChatServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/AbstractDiskHttpData.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker00.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvous/MsgEchoPeerHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/echo/EchoClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/GlobalEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/snoop/HttpSnoopClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/EventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostBodyUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvousBytes/ByteEchoPeerBase.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/oio/AbstractOioChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/FileUpload.java
*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/SnappyFrameDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/server/WebSocketServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledHeapByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/ZlibCodecFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/ChunkedInput.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5PasswordAuthRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker08.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksCmdResponseDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/bytes/ByteEchoClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/CommonsLoggerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultEventLoopGroup.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/InternalLoggerFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DecoderException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandResponse.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpObjectDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/ChunkedNioFile.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpHeaderDateFormat.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/FactorialClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v4/Socks4ServerDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/objectecho/ObjectEchoServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Attribute.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultHttpResponse.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspResponseEncoder.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/stream/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/ssl/NotSslRecordException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostRequestEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5PasswordAuthRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/snoop/HttpSnoopServerHandler.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/SnappyFrameEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/CombinedIterator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/DefaultHttpHeaders.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/Socks5CommandRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspObjectEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/timeout/ReadTimeoutException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker08.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/string/StringDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DatagramChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/file/FileServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/CombinedChannelDuplexHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerProtocolHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspRequestDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/telnet/TelnetServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/DecoderResult.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/example/http/upload/HttpUploadServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/concurrent/SingleThreadEventExecutor.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/oio/AbstractOioMessageChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/bootstrap/AbstractBootstrap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelOutboundHandlerAdapter.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/telnet/TelnetServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/EventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/socksproxy/SocksServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/upload/HttpUploadServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledDirectByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpDataFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/nio/NioServerSocketChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketClientHandshaker13.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/logging/LoggingHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/message/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/ZlibDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/embedded/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/oio/OioDatagramChannel.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/proxy/HexDumpProxy.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/group/ChannelGroupFuture.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/CookieDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/CompatibleMarshallingDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ReplayingDecoderByteBuf.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/spdy/SpdyHttpResponseStreamIdHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/sctp/SctpEchoClientHandler.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/InternetProtocolFamily.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/AbstractByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/qotm/QuoteOfTheMomentClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/Timeout.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpMethod.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelFuture.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/QueryStringEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/UnpooledUnsafeDirectByteBuf.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/string/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/MessageToByteEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/PoolArena.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/DefaultSocks5InitialRequest.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/WebSocketServerHandshaker07.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/TooLongFrameException.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpContentDecompressor.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socksx/v5/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/ChannelInputShutdownEvent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/telnet/TelnetServerInitializer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/ServerSocketChannelConfig.java

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/rendezvous/Config.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/PlatformDependent.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/ContextBoundUnmarshallerProvider.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/FactorialServer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/nio/NioEventLoop.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/securechat/SecureChatClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/worldclock/WorldClockServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/AbstractHttpData.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/MemoryAttribute.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/factorial/FactorialServerHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/DefaultChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/buffer/ByteBuf.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/socks/SocksInitRequestDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/compression/JZlibDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/MessageToMessageDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/ObjectDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/HttpRequestDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/udt/DefaultUdtChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/ReplayingDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/bytes/ByteArrayEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/CompatibleObjectEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-

jar/io/netty/channel/AdaptiveRecvByteBufAllocator.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/socket/DefaultSocketChannelConfig.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/HttpPostStandardRequestDecoder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/uptime/UptimeClientHandler.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/serialization/WeakReferenceMap.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/HashedWheelTimer.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/multipart/DefaultHttpDataFactory.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/package-info.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspRequestEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/ChannelHandlerContext.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/udt/echo/message/MsgEchoClient.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/marshalling/ThreadLocalUnmarshallerProvider.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/rtsp/RtspMethods.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackDynamicTable.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackStaticTable.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackUtil.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackDecoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackHuffmanDecoder.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackHuffmanEncoder.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/HpackHeaderField.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/ocsp/OcspUtils.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/ocsp/OcspClientExample.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/ocsp/OcspRequestBuilder.java
*
 /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/ocsp/OcspServerExample.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/ocsp/Digester.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
//The MIT License
//Permission is hereby granted, free of charge, to any person obtaining a copy
//of this software and associated documentation files (the "Software"), to deal
//to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
//furnished to do so, subject to the following
 conditions:
//The above copyright notice and this permission notice shall be included in
//all copies or substantial portions of the Software.
```

Found in path(s):

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/example/http/websocketx/client/WebSocketClientHandler.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/InternalLogger.java
*
/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/CommonsLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/Log4JLogger.java
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/JdkLogger.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2014 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
/**
    * Set the {@code TCP_MD5SIG} option on the socket. See {@code linux/tcp.h} for more details.
    * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
    * Allowing them being read would mean anyone with access to the channel could get them.

  */
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollServerSocketChannelConfig.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2011 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/sctp/DefaultSctpServerChannelConfig.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/MessageFormatter.java
*

/opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/util/internal/logging/FormattingTuple.java

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http2/Http2ConnectionPrefaceWrittenEvent.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/handler/codec/http/websocketx/Utf8Validator.java
No license file was found, but licenses were detected in source scan.

```
/**
    * Set the {@code TCP_MD5SIG} option on the socket. See {@code linux/tcp.h} for more details.
    * Keys can only be set on, not read to prevent a potential leak, as they are confidential.
    * Allowing them being read would mean anyone with access to the channel could get them.

  */
/**
    * Set the {@code TCP_QUICKACK} option on the socket. See <a
```

```
href="http://linux.die.net/man/7/tcp">TCP_QUICKACK</a>
    * for more details.
    */
```

Found in path(s):
* /opt/cola/permits/175563383_1695324943.6743262/0/netty-all-4-1-11-final-sources-jar/io/netty/channel/epoll/EpollSocketChannelConfig.java

1.125 bcpkix-jdk15on 1.55

1.126 amqp-client 3.6.2

1.127 commons-validator 1.5.1

1.127.1 Available under license :

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership
  of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.
```

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Validator
Copyright 2001-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.128 jackson-jaxrs-base 2.7.4

1.128.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.129 netty 4.1.16

1.129.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too,
but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge

for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary

General Public License therefore permits such linking only if the
entire combination
 fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

 We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

 For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only,
 so we use the Lesser General Public License.

 In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

 Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

 The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

   GNU LESSER GENERAL PUBLIC
 LICENSE
  TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of

this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included
without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately
publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices
stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no
charge to all third parties under the terms of this License.

d) If a facility in the
modified Library refers to a function or a
table of data to be supplied by an application program that uses
the facility, other than as an argument passed when the facility
is invoked, then you must make a good faith effort to ensure that,
in the event an application does not supply such function or
table, the facility still operates, and performs whatever part of
its purpose remains meaningful.

(For example, a function in a library to compute square roots has
a purpose that is entirely well-defined independent of the
application. Therefore, Subsection 2d requires that any
application-supplied function or table used by this function must
be optional: if the application does not supply it, the square
root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works
in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this

License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must
be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered
by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable
source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying
or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you
cannot
distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is

implemented by public license practices.
Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General
Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the
free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR

OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You
should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

The Netty Project
=================

Please visit the Netty web site for more information:

* http://netty.io/

Copyright 2014 The Netty Project

The Netty Project licenses this file to you under the Apache License, version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at:

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This
product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:
   * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified portion of 'Webbit', an event based
WebSocket and HTTP server, which can be obtained at:

 * LICENSE:
   * license/LICENSE.webbit.txt (BSD License)
 * HOMEPAGE:
   * https://github.com/joewalnes/webbit

This product contains a modified portion of 'SLF4J', a simple logging
facade for Java,
which can be obtained at:

 * LICENSE:
   * license/LICENSE.slf4j.txt (MIT License)
 * HOMEPAGE:
   * http://www.slf4j.org/

This product contains a modified portion of 'Apache Harmony', an open source
Java SE, which can be obtained at:

 * LICENSE:
   * license/LICENSE.harmony.txt (Apache License 2.0)
 * HOMEPAGE:
   * http://archive.apache.org/dist/harmony/

This product contains a modified portion of 'jbzip2', a Java bzip2 compression
and decompression library written by Matthew J. Francis. It can be obtained at:

 * LICENSE:
   * license/LICENSE.jbzip2.txt (MIT License)
 * HOMEPAGE:
   * https://code.google.com/p/jbzip2/

This product contains a modified portion of 'libdivsufsort', a C API library to construct
the suffix array and the Burrows-Wheeler transformed string for any input string of
a constant-size alphabet written by Yuta Mori. It can be obtained at:

 * LICENSE:
   * license/LICENSE.libdivsufsort.txt (MIT License)
 * HOMEPAGE:
   * https://github.com/y-256/libdivsufsort

This
 product contains a modified portion of Nitsan Wakart's 'JCTools', Java Concurrency Tools for the JVM,
which can be obtained at:

 * LICENSE:
   * license/LICENSE.jctools.txt (ASL2 License)
 * HOMEPAGE:
   * https://github.com/JCTools/JCTools

This product optionally depends on 'JZlib', a re-implementation of zlib in
pure Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.jzlib.txt (BSD style License)
 * HOMEPAGE:
   * http://www.jcraft.com/jzlib/

This product optionally depends on 'Compress-LZF', a Java library for encoding and
decoding data in LZF format, written by Tatu Saloranta. It can be obtained at:

 * LICENSE:
   * license/LICENSE.compress-lzf.txt (Apache License 2.0)

* HOMEPAGE:
  * https://github.com/ning/compress

This product optionally depends on 'lz4', a LZ4 Java compression
and decompression library written by Adrien Grand. It can be obtained at:

* LICENSE:
  * license/LICENSE.lz4.txt (Apache License 2.0)
*
HOMEPAGE:
  * https://github.com/jpountz/lz4-java

This product optionally depends on 'lzma-java', a LZMA Java compression
and decompression library, which can be obtained at:

* LICENSE:
  * license/LICENSE.lzma-java.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/jponge/lzma-java

This product contains a modified portion of 'jfastlz', a Java port of FastLZ compression
and decompression library written by William Kinney. It can be obtained at:

* LICENSE:
  * license/LICENSE.jfastlz.txt (MIT License)
* HOMEPAGE:
  * https://code.google.com/p/jfastlz/

This product contains a modified portion of and optionally depends on 'Protocol Buffers', Google's data
interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/protobuf

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate
a temporary self-signed X.509 certificate when the JVM does not
provide the
equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'Snappy', a compression library produced
by Google Inc, which can be obtained at:

* LICENSE:
  * license/LICENSE.snappy.txt (New BSD License)
* HOMEPAGE:
  * https://github.com/google/snappy

This product optionally depends on 'JBoss Marshalling', an alternative Java serialization API, which can be obtained at:

* LICENSE:
  * license/LICENSE.jboss-marshalling.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://www.jboss.org/jbossmarshalling

This product optionally depends on 'Caliper', Google's micro-benchmarking framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.caliper.txt (Apache License 2.0)
* HOMEPAGE:
  * https://github.com/google/caliper

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be
obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'Aalto XML', an ultra-high performance non-blocking XML processor, which can be obtained at:

* LICENSE:
  * license/LICENSE.aalto-xml.txt (Apache License 2.0)
* HOMEPAGE:
  * http://wiki.fasterxml.com/AaltoHome

This product contains a modified version of 'HPACK', a Java implementation of the HTTP/2 HPACK algorithm written by Twitter. It can be obtained at:

 * LICENSE:
   * license/LICENSE.hpack.txt (Apache License 2.0)
 * HOMEPAGE:
   * https://github.com/twitter/hpack

This product contains a modified portion of 'Apache Commons Lang', a Java library provides utilities for the java.lang API, which can be obtained at:

 * LICENSE:
   * license/LICENSE.commons-lang.txt (Apache License 2.0)
 * HOMEPAGE:
   * https://commons.apache.org/proper/commons-lang/

The MIT License

Copyright (c) 2009 William Kinney

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity
of all present
and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.
This copy of Aalto XML processor is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/

A copy is also included with both the the downloadable source code package
and jar that contains class bytecodes, as file "ASL 2.0". In both cases,
that file should be located next to this file: in source distribution
the location should be "release-notes/asl"; and in jar "META-INF/"
(BSD License: http://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

* Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the
following disclaimer in the documentation and/or other

materials provided with the distribution.

* Neither the name of the Webbit nor the names of
its contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
Copyright 2011, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2010-2011 Matthew J. Francis and Contributors of the jbzip2 Project

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright 2009-2010 Ning, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Protocol Buffers - Google's data interchange format
Copyright 2013 Google Inc. All rights reserved.
https://developers.google.com/protocol-buffers/

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.
Copyright (c) 2003-2008 Yuta Mori All Rights Reserved.

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,

copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
/*
* Copyright (c) 2004-2007 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,

each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.130 minlog 1.2

1.131 antlr-runtime 4.5.3

1.131.1 Available under license :

THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project uses other open source projects, which are used under the terms
of the following license(s).

.NET Core, .NET Reference Source

mcs/class (from Mono)

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

mcs/class (from Mono)

Copyright (C) 2009 Novell, Inc (http://www.novell.com)

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation
files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
[The "BSD license"]
Copyright (c) 2015 Terence Parr, Sam Harwell
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,

ANTLR Project Contributors Certification of Origin and Rights

All contributors to ANTLR v4 must formally agree to abide by this certificate of origin by signing on the bottom with their github userid, full name, email address (you can obscure your e-mail, but it must be computable by human), and date.

By signing this agreement, you are warranting and representing that you have the right to release code contributions or other content free of any obligations to third parties and are granting Terence Parr and ANTLR project contributors, henceforth referred to as The ANTLR Project, a license to incorporate it into The ANTLR Project tools (such as ANTLRWorks and StringTemplate) or related works under the BSD license. You understand that The ANTLR Project may or may not incorporate your contribution and you warrant and represent the following:

1. I am the creator of all my contributions. I am the author of all contributed work submitted and further warrant and represent that such work is my original creation and I have the right to license it to The ANTLR Project for release under the 3-clause BSD license. I hereby grant The ANTLR Project a nonexclusive, irrevocable, royalty-free, worldwide license to reproduce, distribute, prepare derivative works, and otherwise use this contribution as part of the ANTLR project, associated documentation, books, and tools at no cost to The ANTLR Project.

2. I have the right to submit. This submission does not violate the rights of any person or entity and that I have legal authority over this submission and to make this certification.

3. If I violate another's rights, liability lies with me. I agree to defend, indemnify, and hold The ANTLR Project and ANTLR users harmless from any claim or demand, including reasonable attorney fees, made by any third party due to or arising out of my violation of these terms and conditions or my violation of the rights of another person or entity.

4. I understand

and agree that this project and the contribution are
public and that a record of the contribution (including all
personal information I submit with it, including my sign-off) is
maintained indefinitely and may be redistributed consistent with
this project or the open source license indicated in the file.

I have read this agreement and do so certify by adding my signoff to
the end of the following contributors list.

CONTRIBUTORS:

YYYY/MM/DD, github id, Full name, email
2012/07/12, parrt, Terence Parr, parrt@antlr.org
2012/09/18, sharwell, Sam Harwell, sam@tunnelvisionlabs.com
2012/10/10, stephengaito, Stephen Gaito, stephen@percepitsys.co.uk
2012/11/23, maguro, Alan Cabrera, adc@toolazydogs.com
2013/01/29, metadave, Dave Parfitt, diparfitt@gmail.com
2013/03/06, bkiers, Bart Kiers, bkiers@gmail.com
2013/08/20, cayhorstmann, Cay Horstmann, cay@horstmann.com
2014/03/18, aphyr, Kyle Kingsbury, aphyr@aphyr.com
2014/06/07, ericvergnaud, Eric Vergnaud, eric.vergnaud@wanadoo.fr
2014/07/04,
jimidle, Jim Idle, jimi@Idle.ws
2014/09/04. jeduden, Jan-Eric Duden, jeduden@gmail.com
2014/09/27, petrbel, Petr Blohlvek, antlr@petrbel.cz
2014/10/18, sergiusignacius, Srgio Silva, serge.a.silva@gmail.com
2014/10/26, bdkearns, Brian Kearns, bdkearns@gmail.com
2014/10/27, michaelpj, Michael Peyton Jones, michaelpj@gmail.com
2015/01/29, TomLottermann, Thomas Lottermann, tomlottermann@gmail.com
2015/02/15, pavlo, Pavlo Lysov, pavlikus@gmail.com
2015/03/07, RedTailedHawk, Lawrence Parker, larry@answerrocket.com
2015/04/03, rljacobson, Robert Jacobson, rljacobson@gmail.com
2015/04/06, ojakubcik, Ondrej Jakubcik, ojakubcik@gmail.com
2015/04/29, jszheng, Jinshan Zheng, zheng_js@hotmail.com
2015/05/08, ViceIce, Michael Kriese, michael.kriese@gmx.de
2015/05/09, lkraz, Luke Krasnoff, luke.krasnoff@gmail.com
2015/05/12, Pursuit92, Josh Chase, jcjoshuachase@gmail.com
2015/05/20, peturingi, Ptur Ingi Egilsson, petur@petur.eu
2015/05/27, jcbrinfo, Jean-Christophe Beaupr, jcbrinfo@users.noreply.github.com
2015/06/29,
jvanzyl, Jason van Zyl, jason@takari.io
2015/08/18, krzkaczor, Krzysztof Kaczor, krzysztof@kaczor.io
2015/09/18, worsht, Rajiv Subrahmanyam, rajiv.public@gmail.com
2015/09/24, HSorensen, Henrik Sorensen, henrik.b.sorensen@gmail.com
2015/10/06, brwml, Bryan Wilhelm, bryan.wilhelm@microsoft.com
2015/10/08, fedotovalex, Alex Fedotov, me@alexfedotov.com
2015/10/12, KvanTTT, Ivan Kochurkin, ivan.kochurkin@gmail.com
2015/10/21, martin-probst, Martin Probst, martin-probst@web.de

2015/10/21, hkff, Walid Benghabrit, walid.benghabrit@mines-nantes.fr
2015/11/12, cooperra, Robbie Cooper, cooperra@users.noreply.github.com
2015/11/25, abego, Udo Borkowski, ub@abego.org
2015/12/17, sebadur, Sebastian Badur, sebadur@users.noreply.github.com
2015/12/23, pboyer, Peter Boyer, peter.b.boyer@gmail.com
2015/12/24, dtymon, David Tymon, david.tymon@gmail.com
2016/02/18, reitzig, Raphael Reitzig, reitzig[at]cs.uni-kl.de
2016/03/27, beardlybread, Bradley Steinbacher, bradley.j.steinbacher@gmail.com
2016/03/29,
 msteiger, Martin Steiger, antlr@martin-steiger.de
2016/03/28, gagern, Martin von Gagern, gagern@ma.tum.de

1.132 cassandra-driver-mapping 3.0.2

1.132.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
*      Copyright (C) 2012-2015 DataStax Inc.
*
*   Licensed under the Apache License, Version 2.0 (the "License");
*   you may not use this file except in compliance with the License.
*   You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
*   Unless required by applicable law or agreed to in writing, software
*   distributed under the License is distributed on an "AS IS" BASIS,
*   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*   See the License for the specific language governing permissions and
*   limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Column.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Accessor.java
*
 /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/ReflectionUtils.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/DriverThrowables.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/UDT.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/AccessorInvocationHandler.java

* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Transient.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/AccessorReflectionMapper.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/MethodMapper.java
*
/opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/MapperBoundStatement.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Table.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/AccessorMapper.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Defaults.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/FrozenValue.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/EntityMapper.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Frozen.java
*
/opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Param.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/FrozenKey.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Field.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/Mapper.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/MappedUDTCodec.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/AnnotationParser.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Query.java
*
/opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/TypeMappings.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/PartitionKey.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/MappingManager.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/Result.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/AnnotationChecks.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-

jar/com/datastax/driver/mapping/annotations/ClusteringColumn.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/ColumnMapper.java
*
/opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/Computed.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/QueryType.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/ReflectionMapper.java
* /opt/cola/permits/176115612_1695325478.1612442/0/cassandra-driver-mapping-3-0-2-sources-jar/com/datastax/driver/mapping/annotations/QueryParameters.java

1.133 google-api-client 1.22.0

1.133.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/compute/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/TestUtils.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/compute/MockMetadataServerTransport.java
*
/opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/util/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/DefaultCredentialProvider.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/OAuth2Utils.java

* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/auth/oauth2/MockTokenServerTransport.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/auth/oauth2/package-info.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/SystemEnvironmentProvider.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/CloudShellCredential.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleAuthorizationCodeTokenRequest.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleClientSecrets.java
*
/opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/clientlogin/ClientLoginResponseException.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleBrowserClientRequestUrl.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleCredential.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/json/GoogleJsonError.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/package-info.java

* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/media/MediaHttpUploader.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleAuthorizationCodeRequestUrl.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/json/GoogleJsonResponseException.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/media/MediaHttpUploaderProgressListener.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleOAuthConstants.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleRefreshTokenRequest.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/media/MediaHttpDownloaderProgressListener.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/json/CommonGoogleJsonClientRequestInitializer.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/BatchCallback.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleAuthorizationCodeFlow.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/BatchRequest.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/json/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/json/AbstractGoogleJsonClient.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/json/AbstractGoogleJsonClientRequest.java

* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleIdToken.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleTokenResponse.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/json/JsonBatchCallback.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/json/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/json/GoogleJsonErrorContainer.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/BatchUnparsedResponse.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GoogleIdTokenVerifier.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/media/MediaHttpDownloader.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/javanet/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/apache/GoogleApacheHttpTransport.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/GoogleUtils.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/UnparsedNotification.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-

jar/com/google/api/client/googleapis/notifications/TypedNotificationCallback.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/json/JsonNotificationCallback.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/TypedNotification.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/batch/HttpRequestContent.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/apache/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/util/Utils.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/UnparsedNotificationCallback.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/NotificationUtils.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/notifications/MockUnparsedNotificationCallback.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/notifications/package-info.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/json/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/json/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/AbstractNotification.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/compute/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/json/GoogleJsonResponseExceptionFactoryTesting.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/ResourceStates.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/compute/ComputeCredential.java
*
 /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/media/MediaUploadErrorHandler.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/oauth2/GooglePublicKeysManager.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/notifications/StoredChannel.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/javanet/GoogleNetHttpTransport.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/services/json/MockGoogleJsonClient.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/CommonGoogleClientRequestInitializer.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/AbstractGoogleClientRequest.java
*
/opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/services/json/MockGoogleJsonClientRequest.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/services/MockGoogleClientRequest.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/services/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/GoogleClientRequestInitializer.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/services/json/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/testing/services/MockGoogleClient.java
*
/opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/AbstractGoogleClient.java
No license file was found, but licenses were detected in source scan.

express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/clientlogin/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/MethodOverride.java
*
/opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/media/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/clientlogin/AuthKeyValueParser.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/auth/clientlogin/ClientLogin.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/json/package-info.java
* /opt/cola/permits/176116438_1695327275.1492558/0/google-api-client-1-22-0-sources-jar/com/google/api/client/googleapis/services/package-info.java

1.134 googleapiclientjackson 1.20.0

1.134.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176116596_1695327277.3838375/0/google-http-client-jackson2-1-20-0-sources-zip/com/google/api/client/json/jackson2/JacksonGenerator.java

* /opt/cola/permits/176116596_1695327277.3838375/0/google-http-client-jackson2-1-20-0-sources-zip/com/google/api/client/json/jackson2/package-info.java
* /opt/cola/permits/176116596_1695327277.3838375/0/google-http-client-jackson2-1-20-0-sources-zip/com/google/api/client/json/jackson2/JacksonFactory.java
*
 /opt/cola/permits/176116596_1695327277.3838375/0/google-http-client-jackson2-1-20-0-sources-zip/com/google/api/client/json/jackson2/JacksonParser.java

1.135 googleapiclientjava 1.20.0

1.135.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/176116747_1695325487.7539616/0/google-api-client-java6-1-20-0-sources-jar/com/google/api/client/googleapis/extensions/java6/auth/oauth2/package-info.java
* /opt/cola/permits/176116747_1695325487.7539616/0/google-api-client-java6-1-20-0-sources-jar/com/google/api/client/googleapis/extensions/java6/auth/oauth2/GooglePromptReceiver.java

1.136 google-http-client 1.22.0

1.136.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
```

Found in path(s):
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/BackOff.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Data.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/ExponentialBackOffPolicy.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/AbstractHttpContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/FileContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Value.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/AbstractInputStreamContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ExponentialBackOff.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/BasicAuthentication.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/BackOffPolicy.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/ByteArrayContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/NullValue.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonString.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/UriTemplate.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpStatusCodes.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Types.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpRequestInitializer.java
*

/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/MockHttpUnsuccessfulResponseHandler.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpRequestFactory.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/HttpTesting.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ArrayValueMap.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Objects.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/MemoryDataStoreFactory.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpIOExceptionHandler.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Joiner.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/DataStore.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpBackOffIOExceptionHandler.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/BackOffUtils.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Maps.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/AbstractDataStore.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/DataStoreFactory.java
*

/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/util/MockSleeper.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/json/MockJsonParser.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/webtoken/package-info.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/json/MockJsonGenerator.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/apache/MockHttpClient.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/AbstractDataStoreFactory.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/PemReader.java

*

/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Sleeper.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Charsets.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpBackOffUnsuccessfulResponseHandler.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/FileDataStoreFactory.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Beta.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Sets.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/package-info.java

*

/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ByteStreams.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Collections2.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Preconditions.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/SecurityUtils.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonPolymorphicTypeMap.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/json/package-info.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/util/TestableByteArrayInputStream.java

*

/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/util/MockBackOff.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-

jar/com/google/api/client/testing/util/TestableByteArrayOutputStream.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/AbstractMemoryDataStore.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Throwables.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/MultipartContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/apache/SSLSocketFactoryExtension.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/json/MockJsonFactory.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Lists.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/apache/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Strings.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/store/DataStoreUtils.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/util/SecurityTestUtils.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/NanoClock.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2010 Google Inc.J
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except

```
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/apache/HttpExtensionMethod.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpMethods.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/json/webtoken/package-info.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
```

Found in path(s):
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/javanet/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpResponseException.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpTransport.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/apache/ApacheHttpResponse.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ClassInfo.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonToken.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/apache/ContentEntity.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/MockLowLevelHttpResponse.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpContent.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpUnsuccessfulResponseHandler.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/escape/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpResponseInterceptor.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/javanet/NetHttpTransport.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpHeaders.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpExecuteInterceptor.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/DataMap.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpRequest.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/rpc2/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-

jar/com/google/api/client/http/apache/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/javanet/NetHttpResponse.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/escape/UnicodeEscaper.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/InputStreamContent.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/CustomizeJsonParser.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/GenericData.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/apache/ApacheHttpRequest.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/escape/Platform.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/Json.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonParser.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/LowLevelHttpResponse.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonGenerator.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/rpc2/JsonRpcRequest.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpResponse.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/GenericJson.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/javanet/NetHttpRequest.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/MockLowLevelHttpRequest.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ArrayMap.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/LoggingStreamingContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/escape/Escaper.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/MockHttpTransport.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/UrlEncodedParser.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/MockHttpContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Key.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/json/JsonHttpContent.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/DateTime.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/apache/ApacheHttpTransport.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/escape/PercentEscaper.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/LowLevelHttpRequest.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/UrlEncodedContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/GenericUrl.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/json/package-info.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/FieldInfo.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/escape/CharEscapers.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpEncoding.java

* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/FixedClock.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/StringUtils.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpEncodingStreamingContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/webtoken/JsonWebSignature.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/HttpMediaType.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/webtoken/JsonWebToken.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/LoggingOutputStream.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/util/LogRecordingHandler.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/javanet/MockHttpURLConnection.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/LoggingByteArrayOutputStream.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ByteArrayStreamingContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/http/GZipEncoding.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Base64.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ByteCountingOutputStream.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/StreamingContent.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/Clock.java
*
 /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/json/JsonObjectParser.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/util/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/http/javanet/package-info.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/testing/json/webtoken/TestCertificates.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/IOUtils.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/SslUtils.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-

jar/com/google/api/client/http/EmptyContent.java
*
/opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/ObjectParser.java
* /opt/cola/permits/176117839_1695327279.2871747/0/google-http-client-1-22-0-sources-jar/com/google/api/client/util/LoggingInputStream.java

1.137 google-oauth-client 1.22.0

1.137.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/StoredCredential.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/openidconnect/IdToken.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/PasswordTokenRequest.java
*
/opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/ClientCredentialsTokenRequest.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/DataStoreCredentialRefreshListener.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/openidconnect/IdTokenVerifier.java
No license file was found, but licenses were detected in source scan.

* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthRsaSigner.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthGetTemporaryToken.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/AbstractOAuthGetToken.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthCredentialsResponse.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthSigner.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthAuthorizeTemporaryTokenUrl.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthCallbackUrl.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthGetAccessToken.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthHmacSigner.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/OAuthParameters.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/CredentialRefreshListener.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/openidconnect/IdTokenResponse.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/AuthorizationCodeFlow.java
*
 /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/CredentialStore.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/Credential.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/RefreshTokenRequest.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/BearerToken.java
*
 /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/AuthorizationCodeRequestUrl.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/TokenRequest.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/package-info.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/TokenResponse.java

* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/TokenErrorResponse.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/openidconnect/package-info.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/ClientParametersAuthentication.java
*
 /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/BrowserClientRequestUrl.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/AuthorizationCodeTokenRequest.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/AuthorizationCodeResponseUrl.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/CredentialStoreRefreshListener.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/TokenResponseException.java
* /opt/cola/permits/176118237_1695327284.4814794/0/google-oauth-client-1-22-0-sources-jar/com/google/api/client/auth/oauth2/AuthorizationRequestUrl.java

1.138 groovy-console 2.4.6

1.138.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--

  Licensed to the Apache Software Foundation (ASF) under one
  or more contributor license agreements.  See the NOTICE file
  distributed with this work for additional information
  regarding copyright ownership.  The ASF licenses this file
  to you under the Apache License, Version 2.0 (the
  "License"); you may not use this file except in compliance
  with the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
  software distributed under the License is distributed on an
  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
  KIND, either express or implied.  See the License for the
  specific language governing permissions and limitations
  under the License.

-->
```

Found in path(s):

* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/package.html
*
 /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/swingui/package.html
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/package.html
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

These icons are from the famfamfam.com silk icons set
http://www.famfamfam.com/lab/icons/silk/

Licensed under
 the Creative Commons Attribution Licence v2.5
http://creativecommons.org/licenses/by/2.5/

Found in path(s):
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/icons/credits.txt
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*

Found in path(s):
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/ConsoleSupport.java
*
 /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/WindowsDefaults.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/Console.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/HistoryRecord.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/BasicToolBar.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/GroovyFilter.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/ConsoleTextEditor.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/swingui/AstNodeToScriptAdapter.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/swingui/AstBrowser.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/FindReplaceUtility.java
*
 /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/TextNode.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/BasicStatusBar.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/ConsoleActions.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/StructuredSyntaxHandler.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/ConsoleApplet.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/TextEditor.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/AutoIndentAction.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-

jar/groovy/ui/view/MacOSXMenuBar.groovy
*
/opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/SystemOutputInterceptor.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/StructuredSyntaxDocumentFilter.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/ConsoleIvyPlugin.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/StructuredSyntaxResources.java
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/OutputTransforms.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/swingui/ScriptToTreeNodeAdapter.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/TextTreeNodeMaker.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/BasicMenuBar.groovy
*
/opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/Defaults.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/MacOSXDefaults.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/swingui/AstBrowserProperties.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/BasicContentPane.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/inspect/swingui/ObjectBrowser.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/view/GTKDefaults.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/ConsoleView.groovy
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/text/TextUndoManager.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176118858_1695327288.2340786/0/groovy-console-2-4-6-sources-jar/groovy/ui/InteractiveShell.properties

1.139 groovy-groovysh 2.4.6

1.139.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Adapted from JLine which has the following license
*
*  Copyright (c) 2002-2012, the original author or authors.
*  This software is distributable under the BSD license. See the terms of the
*  BSD license in the documentation provided with this software.
*
*    http://www.opensource.org/licenses/bsd-license.php
*
* Subsequent modifications by the Groovy community have been done under the Apache License v2:
*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
*  distributed with this work for additional information
*  regarding copyright ownership.  The ASF licenses this file
*  to you under the Apache License, Version 2.0 (the
*  "License"); you may not use this file except in compliance
*  with the License.  You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required
 by applicable law or agreed to in writing,
*  software distributed under the License is distributed on an
*  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
*  KIND, either express or implied.  See the License for the
*  specific language governing permissions and limitations
*  under the License.
*/
```

Found in path(s):

* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/FileNameCompleter.groovy

No license file was found, but licenses were detected in source scan.

```
#  to you under the Apache License, Version 2.0 (the
#  "License"); you may not use this file except in compliance
#  with the License.  You may obtain a copy of the License at
#    http://www.apache.org/licenses/LICENSE-2.0
#  software distributed under the License is distributed on an
```

Found in path(s):

* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/RecordCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/DisplayCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/EditCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/PurgeCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ShowCommand.properties
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Main.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/DocCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/HelpCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ShadowCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ImportCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/LoadCommand.properties
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/HistoryCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/RegisterCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/AliasCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/CommandSupport.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/InspectCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/SetCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ExitCommand.properties
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/CommandAlias.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/SaveCommand.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Groovysh.properties
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ClearCommand.properties

No license file was found, but licenses were detected in source scan.

```
<!--

    Licensed to the Apache Software Foundation (ASF) under one
    or more contributor license agreements.  See the NOTICE file
    distributed with this work for additional information
    regarding copyright ownership.  The ASF licenses this file
    to you under the Apache License, Version 2.0 (the
    "License"); you may not use this file except in compliance
    with the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing,
    software distributed under the License is distributed on an
    "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
    KIND, either express or implied.  See the License for the
    specific language governing permissions and limitations
    under the License.

-->
```

Found in path(s):
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/package.html
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/package.html
No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
*  distributed with this work for additional information
*  regarding copyright ownership.  The ASF licenses this file
*  to you under the Apache License, Version 2.0 (the
*  "License"); you may not use this file except in compliance
*  with the License.  You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing,
*  software distributed under the License is distributed on an
*  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
*  KIND, either express or implied.  See the License for the
*  specific language governing permissions and limitations
*  under the License.
```

*/

Found in path(s):
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/BufferManager.groovy
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/DocCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/CustomClassSyntaxCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/XmlCommandRegistrar.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/HelpFormatter.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Groovysh.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/SetCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/DisplayCommand.groovy
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/HelpCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ClearCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Interpreter.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/NoExitSecurityManager.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/InfixKeywordSyntaxCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/CurlyCountingGroovyLexer.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/HistoryCommand.groovy
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/PurgeCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/AnsiDetector.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/ReflectionCompletionCandidate.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ShadowCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/NavigablePropertiesCompleter.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/SimpleCompletor.groovy

* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/InteractiveShellRunner.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/PatchedCandidateListCompletionHandler.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/KeywordSyntaxCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Parser.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/DefaultCommandsRegistrar.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/ScriptVariableAnalyzer.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/WrappedInputStream.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ImportCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/ComplexCommandSupport.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/ShellRunner.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/RecordCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/LoadCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/EditCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/SaveCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/IdentifierCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/ImportsSyntaxCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/StricterArgumentCompleter.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/VariableSyntaxCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/CommandNameCompleter.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/PackageHelperImpl.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/PatchedConsoleReader.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/JAnsiHelper.groovy

* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/AliasCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/CommandAlias.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ExitCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/GroovySyntaxCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Shell.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/ExitNotification.groovy
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/RegisterCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/ShowCommand.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/completion/ReflectionCompletor.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/CommandArgumentParser.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/CommandException.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/util/PackageHelper.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Main.groovy
*
 /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/CommandRegistry.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/CommandSupport.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/Command.groovy
* /opt/cola/permits/176118970_1695325504.296089/0/groovy-groovysh-2-4-6-sources-jar/org/codehaus/groovy/tools/shell/commands/InspectCommand.groovy

1.140 groovy-json 2.4.6

1.140.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
*  distributed with this work for additional information
*  regarding copyright ownership.  The ASF licenses this file
```

Found in path(s):
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Charsets.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonSlurperClassic.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonOutput.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/StringEscapeUtils.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/CharBuf.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/CharScanner.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonLexer.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ValueList.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/CacheType.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Dates.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonTokenType.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/JsonParserLax.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Byt.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonDelegate.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonSlurper.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonException.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonToken.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonParser.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/IO.java
*

/opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/JsonFastParser.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/MapItemValue.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Value.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/NumberValue.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/LazyMap.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ReaderCharacterSource.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Exceptions.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ValueMapImpl.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/CharacterSource.java
*
/opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonBuilder.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/BaseJsonParser.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/JsonParserCharArray.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Type.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ValueMap.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/CharSequenceValue.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/LazyValueMap.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/JsonParserUsingCharacterSource.java
*
/opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/StreamingJsonBuilder.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/FastStringUtils.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ValueContainer.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Chr.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/SimpleCache.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ArrayUtils.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/Sys.java
* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-

jar/groovy/json/internal/Cache.java

*

/opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/JsonParserType.java

* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/ByteScanner.java

* /opt/cola/permits/176119082_1695325507.070628/0/groovy-json-2-4-6-sources-jar/groovy/json/internal/JsonStringDecoder.java

1.141 groovyjsr 2.4.6

1.141.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176119194_1695325509.6919262/0/groovy-jsr223-2-4-6-sources-jar/org/codehaus/groovy/jsr223/GroovyScriptEngineImpl.java
* /opt/cola/permits/176119194_1695325509.6919262/0/groovy-jsr223-2-4-6-sources-jar/org/codehaus/groovy/jsr223/GroovyCompiledScript.java
*
/opt/cola/permits/176119194_1695325509.6919262/0/groovy-jsr223-2-4-6-sources-jar/org/codehaus/groovy/jsr223/GroovyScriptEngineFactory.java
No license file was found, but licenses were detected in source scan.

org.codehaus.groovy.jsr223.GroovyScriptEngineFactory

Found in path(s):
* /opt/cola/permits/176119194_1695325509.6919262/0/groovy-jsr223-2-4-6-sources-jar/META-INF/services/javax.script.ScriptEngineFactory
No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
*  distributed with this work for additional information
*  regarding copyright ownership.  The ASF licenses this file
*  to you under the Apache License, Version 2.0 (the
*  "License"); you may not use this file except in compliance
*  with the License.  You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing,
*  software distributed under the License is distributed on an
*  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
*  KIND, either express or implied.  See the License for the
*  specific language governing permissions and limitations
*  under the License.
*/
```

Found in path(s):
* /opt/cola/permits/176119194_1695325509.6919262/0/groovy-jsr223-2-4-6-sources-jar/org/codehaus/groovy/jsr223/ScriptExtensions.java
*
/opt/cola/permits/176119194_1695325509.6919262/0/groovy-jsr223-2-4-6-sources-jar/org/codehaus/groovy/jsr223/ScriptStaticExtensions.java

1.142 groovy-swing 2.4.6

1.142.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
*  distributed with this work for additional information
*  regarding copyright ownership.  The ASF licenses this file
*  to you under the Apache License, Version 2.0 (the
*  "License"); you may not use this file except in compliance
*  with the License.  You may obtain a copy of the License at
```

Found in path(s):
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JScrollBarProperties.java
*
 /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JTableProperties.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/AbstractButtonProperties.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/GridBagFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/DefaultTableColumn.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BoxFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/MapFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/TableFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ActionFactory.groovy
*
 /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/RendererFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/CompoundBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/EventTriggerBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/TableLayoutRow.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/runtime/SwingGroovyMethods.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/ClosureRenderer.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/WidgetFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/BindPath.java

*

/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/ClosureSourceBinding.java

* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/ValueHolder.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/EtchedBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ImageIconFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JSpinnerProperties.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/TitledBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/PropertyPathFullBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/FullBinding.java

*

/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/DefaultTableModel.java

* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/AbstractFullBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/SwingTimerTriggerBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/PropertyModel.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BevelBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/TableLayoutFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/LineBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JTextComponentProperties.java

*

/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/PropertyBinding.java

* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/DefaultAction.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BoxLayoutFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/EmptyBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BeanFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ColumnFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-

jar/groovy/swing/factory/ComponentFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/ListWrapperListModel.java
*
/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/TriggerBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BindProxyFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/WindowFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ButtonGroupFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JComboBoxProperties.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ScrollPaneFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/TargetBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/ClosureModel.java
*
/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/TableLayout.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/BindingProxy.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/inspect/swingui/TableMap.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/TabbedPaneFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BindGroupFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/BindingUpdatable.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/NestedValueModel.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ListFactory.groovy
*
/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JComponentProperties.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/LayoutFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/CollectionFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/BindFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/AbstractSyntheticMetaMethods.groovy

* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/RootPaneContainerFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/DialogFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/MutualPropertyBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/RichActionWidgetFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/TableModelFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JComboBoxMetaMethods.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ComboBoxFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/ValueModel.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JListMetaMethods.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/FormattedTextFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/ColumnModelFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/SeparatorFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/AggregateBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/InternalFrameFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/SplitPaneFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/PropertyChangeProxyTargetBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/MatteBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/SourceBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/ClosureTriggerBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/LookAndFeelHelper.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/TextArgWidgetFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/TableLayoutCell.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-

jar/groovy/swing/factory/CellEditorFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/SwingBorderFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JTableMetaMethods.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JSliderProperties.java
*
/opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/FrameFactory.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/SwingBuilder.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/ClosureCellEditor.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/JListProperties.groovy
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/FormModel.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/AbstractSyntheticBinding.java
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/inspect/swingui/TableSorter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/binding/package.html

*
 /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/package.html
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/org/codehaus/groovy/binding/package.html
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/factory/package.html
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/model/package.html
* /opt/cola/permits/176119306_1695327290.345556/0/groovy-swing-2-4-6-sources-jar/groovy/swing/impl/package.html

1.143 groovy-templates 2.4.6

1.143.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--

  Licensed to the Apache Software Foundation (ASF) under one
  or more contributor license agreements.  See the NOTICE file
  distributed with this work for additional information
  regarding copyright ownership.  The ASF licenses this file
  to you under the Apache License, Version 2.0 (the
  "License"); you may not use this file except in compliance
  with the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
  software distributed under the License is distributed on an
  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
  KIND, either express or implied.  See the License for the
  specific language governing permissions and limitations
  under the License.

-->
```

Found in path(s):
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/package.html
*
 /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/package.html
No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
```

Found in path(s):
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/TemplateResolver.java
*
/opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/DelegatingIndentWriter.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/SimpleTemplateEngine.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/TemplateExecutionException.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/GStringTemplateEngine.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/TemplateParseException.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/MarkupTemplateTypeCheckingExtension.groovy
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/TemplateEngine.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/MarkupTemplateEngine.java
*
/opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/TemplateConfiguration.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/XmlTemplateEngine.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/TemplateASTTransformer.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/Template.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/BaseTemplate.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-

jar/groovy/text/markup/AutoNewLineTransformer.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/StreamingTemplateEngine.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/IncludeType.java
*
 /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/MarkupBuilderCodeTransformer.java
* /opt/cola/permits/176119418_1695327293.1560552/0/groovy-templates-2-4-6-sources-jar/groovy/text/markup/TagLibAdapter.groovy

1.144 groovy-xml 2.4.6

1.144.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--

  Licensed to the Apache Software Foundation (ASF) under one
  or more contributor license agreements.  See the NOTICE file
  distributed with this work for additional information
  regarding copyright ownership.  The ASF licenses this file
  to you under the Apache License, Version 2.0 (the
  "License"); you may not use this file except in compliance
  with the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing,
  software distributed under the License is distributed on an
  "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
  KIND, either express or implied.  See the License for the
  specific language governing permissions and limitations
  under the License.

-->
```

Found in path(s):
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/package.html
*
 /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/dom/package.html
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/streamingmarkupsupport/package.html
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/package.html
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

Found in path(s):
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/NamespaceBuilder.java
*
/opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/NodeIterator.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/SAXBuilder.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/GPathResult.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/streamingmarkupsupport/BaseMarkupBuilder.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/XmlParser.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/StreamingSAXBuilder.groovy
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/MarkupBuilder.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/org/codehaus/groovy/runtime/XmlGroovyMethods.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/FilteredNodeChildren.java
*
/opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/NamespaceBuilderSupport.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/NodeChild.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/Entity.groovy
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/NamespaceAwareHashMap.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-

jar/groovy/util/slurpersupport/Node.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/FactorySupport.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/NodeChildren.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/StreamingDOMBuilder.groovy
*
/opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/streamingmarkupsupport/AbstractStreamingBuilder.groovy
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/XmlNodePrinter.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/Namespace.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/ReplacementNode.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/org/codehaus/groovy/tools/xml/DomToGroovy.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/dom/DOMCategory.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/XmlUtil.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/Attributes.java
*
/opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/streamingmarkupsupport/Builder.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/StaxBuilder.groovy
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/streamingmarkupsupport/StreamingMarkupWriter.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/Attribute.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/FilteredAttributes.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/DOMBuilder.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/StreamingMarkupBuilder.groovy
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/NoChildren.java
*
/opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/xml/MarkupBuilderHelper.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/XmlSlurper.java
* /opt/cola/permits/176119530_1695325518.5244741/0/groovy-xml-2-4-6-sources-jar/groovy/util/slurpersupport/NodeParents.java

1.145 grpc-okhttp 0.12.0

1.145.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2012 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/Settings.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):

* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/FrameWriter.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/FrameReader.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2015, Google Inc. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
*    * Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*    * Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following disclaimer
* in the documentation and/or other materials provided with the
* distribution.
*
*    * Neither the name of Google Inc. nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED
 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpChannelProvider.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpWritableBuffer.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/package-info.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpWritableBufferAllocator.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/NegotiationType.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpProtocolNegotiator.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/Util.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014, Google Inc. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
*   * Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*   * Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following disclaimer
* in the documentation and/or other materials provided with the
* distribution.
*
*   * Neither the name of Google Inc. nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED
 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/Utils.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/AsyncFrameWriter.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpReadableBuffer.java
*
 /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/Headers.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OutboundFlowController.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpClientStream.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpSettingsUtil.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpClientTransport.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpTlsUpgrader.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/OkHttpChannelBuilder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/Huffman.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2013 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/HeadersMode.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/Http2.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/Variant.java
*
 /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/Hpack.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/framed/ErrorCode.java
No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one or more
*  contributor license agreements.  See the NOTICE file distributed with
*  this work for additional information regarding copyright ownership.
*  The ASF licenses this file to You under the Apache License, Version 2.0
*  (the "License"); you may not use this file except in compliance with
*  the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing, software
*  distributed under the License is distributed on an "AS IS" BASIS,
*  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
*  See the License for the specific language governing permissions and
*  limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/DistinguishedNameParser.java
*
 /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/OkHostnameVerifier.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/OptionalMethod.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/ConnectionSpec.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/Protocol.java
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/CipherSuite.java
*
 /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/TlsVersion.java
No license file was found, but licenses were detected in source scan.

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176120773_1695325521.1790774/0/grpc-okhttp-0-12-0-sources-jar/io/grpc/okhttp/internal/Platform.java

1.146 grpc-protobuf 0.12.0

1.146.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014, Google Inc. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
*   * Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*   * Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following disclaimer
* in the documentation and/or other materials provided with the
* distribution.
*
*   * Neither the name of Google Inc. nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

Found in path(s):
* /opt/cola/permits/176120885_1695325523.7385712/0/grpc-protobuf-0-12-0-sources-jar/io/grpc/protobuf/ProtoInputStream.java
* /opt/cola/permits/176120885_1695325523.7385712/0/grpc-protobuf-0-12-0-sources-jar/io/grpc/protobuf/ProtoUtils.java

1.147 grpc-protobuf-nano 0.12.0

1.147.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014, Google Inc. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are
* met:
*
*   * Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*   * Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following disclaimer
* in the documentation and/or other materials provided with the
* distribution.
*
*   * Neither the name of Google Inc. nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED
 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
* OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/176121111_1695325526.4143646/0/grpc-protobuf-nano-0-12-0-sources-

jar/io/grpc/protobuf/nano/NanoProtoInputStream.java
* /opt/cola/permits/176121111_1695325526.4143646/0/grpc-protobuf-nano-0-12-0-sources-jar/io/grpc/protobuf/nano/NanoUtils.java
* /opt/cola/permits/176121111_1695325526.4143646/0/grpc-protobuf-nano-0-12-0-sources-jar/io/grpc/protobuf/nano/MessageNanoFactory.java

1.148 ivy 2.3.0

1.148.1 Available under license :

Apache Ivy (TM)
Copyright 2007-2013 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

Portions of Ivy were originally developed by
Jayasoft SARL (http://www.jayasoft.fr/)
and are licensed to the Apache Software Foundation under the
"Software Grant License Agreement"

SSH and SFTP support is provided by the JCraft JSch package,
which is open source software, available under
the terms of a BSD style license.
The original software and related information is available
at http://www.jcraft.com/jsch/.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

----

License for JCraft JSch package

License for jQuery

1.149 jackson 2.7.2

1.149.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.150 javax-json 1.0.4

1.150.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU
General Public License Version 2 only ("GPL") or the Common Development
and Distribution License("CDDL") (collectively, the "License"). You
may not use this file except in compliance with the License. You can
obtain a copy of the License at
https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
or packager/legal/LICENSE.txt. See the License for the specific
language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each
file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath"
exception as provided by Oracle in the GPL Version 2 section of the License
file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields
enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license." If you don't indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above. However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if
the new code is made subject to such option by the copyright
holder.

-->

Found in path(s):

* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/META-

INF/maven/org.glassfish/javax.json/pom.xml
No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
```

Found in path(s):
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonPrettyGeneratorImpl.java

* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonParserImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonArrayBuilderImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/UnicodeDetectingInputStream.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonWriter.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonParsingException.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/spi/package-info.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonGeneratorFactoryImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonReader.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonObjectBuilderImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonTokenizer.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/package-info.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonGenerationException.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonGeneratorImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonProviderImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonStructureParser.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonStructure.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonParserFactoryImpl.java

No license file was found, but licenses were detected in source scan.

*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/Json.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonArray.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonValue.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonParser.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonString.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/package-info.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/spi/JsonProvider.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonParserFactory.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonGenerator.java
*
/opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonObject.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonGeneratorFactory.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonException.java

* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonNumber.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonBuilderFactoryImpl.java

* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonReaderFactoryImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonNumberImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonWriterFactory.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonReaderFactory.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonStringImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonReaderImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonLocationImpl.java
*
 /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonBuilderFactory.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonMessages.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonWriterFactoryImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/stream/JsonLocation.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/BufferPoolImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/api/BufferPool.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonObjectBuilder.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/org/glassfish/json/JsonWriterImpl.java
* /opt/cola/permits/176121895_1695325534.5609844/0/javax-json-1-0-4-sources-zip/javax/json/JsonArrayBuilder.java

1.151 javax-json-api 1.0

1.151.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonObjectBuilder.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonBuilderFactory.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonLocation.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonWriterFactory.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonReaderFactory.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonArrayBuilder.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonWriter.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonReader.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/spi/package-

info.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/package-info.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonParsingException.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonGenerationException.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonStructure.java

No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2011-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made

subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonArray.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonNumber.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/package-info.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonException.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonObject.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonString.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/JsonValue.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonGeneratorFactory.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonParser.java
*
/opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonParserFactory.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/Json.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/stream/JsonGenerator.java
* /opt/cola/permits/176122115_1695325537.75786/0/javax-json-api-1-0-sources-zip/javax/json/spi/JsonProvider.java

1.152 jbcrypt 0.4

1.152.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Permission to use, copy, modify, and distribute this software for any
// purpose with or without fee is hereby granted, provided that the above
// copyright notice and this permission notice appear in all copies.

Found in path(s):
* /opt/cola/permits/176122232_1695327297.8240366/0/jbcrypt-0-4-sources-jar/org/mindrot/jbcrypt/BCrypt.java

1.153 jcabi-log 0.14

1.153.1 Available under license :

No license file was found, but licenses were detected in source scan.

/**
* Copyright (c) 2012-2014, jcabi.com
* All rights reserved.
*

* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met: 1) Redistributions of source code must retain the above
* copyright notice, this list of conditions and the following
* disclaimer. 2) Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following
* disclaimer in the documentation and/or other materials provided
* with the distribution. 3) Neither the name of the jcabi.com nor
* the names of its contributors may be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
* NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND
* FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
* THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/DecorException.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/VerboseThreads.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/DomDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/ExceptionDecor.java
*
/opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/TextDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/VerboseRunnable.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/MulticolorLayout.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/ListDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/ObjectDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/TypeDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/NanoDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/DecorsManager.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/SizeDecor.java

*
/opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/VerboseProcess.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/PreFormatter.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/Logger.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/package-info.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/MsDecor.java
* /opt/cola/permits/176122358_1695327299.4885783/0/jcabi-log-0-14-sources-jar/com/jcabi/log/SecretDecor.java

1.154 jcabi-manifests 1.1

1.154.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-

jar/com/jcabi/manifests/FilesMfs.java
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/ServletMfs.java
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/Mfs.java
*
/opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/ClasspathMfs.java
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/Manifests.java
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/StreamsMfs.java
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/MfMap.java
* /opt/cola/permits/176122456_1695325546.5782518/0/jcabi-manifests-1-1-sources-jar/com/jcabi/manifests/package-info.java

1.155 jline 2.12

1.155.1 Available under license :

No license file was found, but licenses were detected in source scan.

# This software is distributable under the BSD license. See the terms of the
# BSD license in the documentation provided with this software.

Found in path(s):
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/CandidateListCompletionHandler.properties
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2002-2012, the original author or authors.
*
* This software is distributable under the BSD license. See the terms of the
* BSD license in the documentation provided with this software.
*
* http://www.opensource.org/licenses/bsd-license.php
*/

Found in path(s):
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/package-info.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/InputStreamReader.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/Operation.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/UnixTerminal.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/NullCompleter.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-

jar/jline/console/completer/EnumCompleter.java
*
/opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/history/package-info.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/ShutdownHooks.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/AggregateCompleter.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/internal/ConsoleRunner.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/Log.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/Terminal.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/Nullable.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/UserInterruptException.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/UnsupportedTerminal.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/TerminalFactory.java
*
/opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/package-info.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/FileNameCompleter.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/history/FileHistory.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/AnsiWindowsTerminal.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/TerminalLineSettings.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/TestAccessible.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/CursorBuffer.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/NoInterruptUnixTerminal.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/package-info.java
*
/opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/CompletionHandler.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/package-info.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/history/History.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/ConsoleReader.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/CandidateListCompletionHandler.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/history/MemoryHistory.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/internal/ConsoleReaderInputStream.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/WindowsTerminal.java
*
/opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/ArgumentCompleter.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/StringsCompleter.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/Urls.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/TerminalSupport.java

* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/KeyMap.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/NonBlockingInputStream.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/Preconditions.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/internal/Configuration.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/completer/Completer.java
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/history/PersistentHistory.java
*
 /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/ConsoleKeys.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176122680_1695325551.5304315/0/jline-2-12-sources-jar/jline/console/KillRing.java

1.156 json-java 20160810

1.156.1 Available under license :

FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.157 minlog 1.3.0

1.158 okhttp 2.5.0

1.158.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/Huffman.java
No license file was found, but licenses were detected in source scan.

* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/TlsVersion.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/PushObserver.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/CipherSuite.java
*
 /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/ConnectionSpec.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Credentials.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/MultipartBuilder.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/CertificatePinner.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/FormEncodingBuilder.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/RealResponseBody.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/HttpMethod.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Challenge.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/Internal.java
*
 /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/Version.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Interceptor.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Callback.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Protocol.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/RequestBody.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Call.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/CacheRequest.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/ResponseBody.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Android Open Source Project
*

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/RetryableSink.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Cache.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Route.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Request.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Dispatcher.java
*
/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/NamedRunnable.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/ErrorCode.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Authenticator.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-

jar/com/squareup/okhttp/internal/framed/Hpack.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/RouteDatabase.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/MediaType.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/InternalCache.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/Http2.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/NameValueBlockReader.java
*
/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/HeadersMode.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Response.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/Variant.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Handshake.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/AuthenticatorAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/ConnectionSpecSelector.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/io/FileSystem.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/RouteException.java
*

/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/RequestException.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/HttpUrl.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/OptionalMethod.java
*
/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Headers.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/tls/DistinguishedNameParser.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/tls/OkHostnameVerifier.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/ConnectionPool.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Connection.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/HttpEngine.java
No license file was found, but licenses were detected in source scan.

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/FramedStream.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/DiskLruCache.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/FramedConnection.java
*
/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/FrameReader.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/FrameWriter.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/IncomingStreamHandler.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/HeaderParser.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/HttpDate.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/Spdy3.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 Square, Inc.
* Copyright (C) 2012 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-

jar/com/squareup/okhttp/internal/Platform.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/Settings.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/RouteSelector.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/framed/Ping.java
*
/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/OkHttpClient.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/Network.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/Transport.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/HttpTransport.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/Util.java
*
/opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/HttpConnection.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/internal/http/FramedTransport.java
* /opt/cola/permits/176123593_1695325559.659631/0/okhttp-2-5-0-sources-jar/com/squareup/okhttp/Address.java

1.159 everit-json-schema 1.4.1

1.159.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

Copyright (C) 2011 Everit Kft. (http://www.everit.org)

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-->

Found in path(s):
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/META-INF/maven/org.everit.json/org.everit.json.schema/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Everit Kft. (http://www.everit.org)
*
* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/ValidationException.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/CombinedSchema.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/Schema.java
*
/opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/SchemaLoader.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/ObjectComparator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/DateTimeFormatValidator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/IPAddressValidator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/IPV6Validator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/FormatValidator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/NullSchema.java
*
/opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/URIFormatValidator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/HostnameFormatValidator.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/SchemaClient.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/EnumSchema.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/BooleanSchema.java
* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/internal/TypeBasedMultiplexer.java
*

/opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/internal/JSONPointer.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/ArraySchema.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/EmailFormatValidator.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/internal/IPV4Validator.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/NotSchema.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/ObjectSchema.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/EmptySchema.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/StringSchema.java

*

/opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/internal/DefaultSchemaClient.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/internal/ResolutionScopeChangeListener.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/SchemaException.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/ReferenceSchema.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/loader/internal/ReferenceResolver.java

* /opt/cola/permits/176123817_1695327302.5521042/0/org-everit-json-schema-1-4-1-sources-2-jar/org/everit/json/schema/NumberSchema.java

1.160 protobuf-javanano 3.0.0-alpha-4

1.160.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/MapFactories.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/InvalidProtocolBufferNanoException.java
*
 /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/CodedOutputByteBufferNano.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/UnknownFieldData.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/MessageNano.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/Extension.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/ExtendableMessageNano.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/MessageNanoPrinter.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/WireFormatNano.java
*
 /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/CodedInputByteBufferNano.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/FieldArray.java
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/FieldData.java
No license file was found, but licenses were detected in source scan.

// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//   * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/176125075_1695325567.2902658/0/protobuf-javanano-3-0-0-alpha-4-sources-jar/com/google/protobuf/nano/InternalNano.java

1.161 reflectasm 1.10.1

1.161.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176125285_1695325570.174006/0/reflectasm-1-10-1-sources-jar/com/esotericsoftware/reflectasm/AccessClassLoader.java
* /opt/cola/permits/176125285_1695325570.174006/0/reflectasm-1-10-1-sources-jar/com/esotericsoftware/reflectasm/ConstructorAccess.java

* /opt/cola/permits/176125285_1695325570.174006/0/reflectasm-1-10-1-sources-jar/com/esotericsoftware/reflectasm/FieldAccess.java
*
 /opt/cola/permits/176125285_1695325570.174006/0/reflectasm-1-10-1-sources-jar/com/esotericsoftware/reflectasm/MethodAccess.java
* /opt/cola/permits/176125285_1695325570.174006/0/reflectasm-1-10-1-sources-jar/com/esotericsoftware/reflectasm/PublicConstructorAccess.java

1.162 util 1.4.5

1.162.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/HttpTransportFactory.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/RetryHttpInitializer.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/LoggingMediaHttpUploaderProgressListener.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/ApiErrorExtractor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/ChainingHttpRequestInitializer.java

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright 2013 Google Inc. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/ClientRequestHelper.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/CredentialFactory.java

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright 2014 Google Inc. All Rights Reserved.
*
*  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
*  http://www.apache.org/licenses/LICENSE-2.0
*
*  Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/CredentialConfiguration.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/EntriesCredentialConfiguration.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/AbstractGoogleAsyncWriteChannel.java
*

/opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/ExceptionToIoExceptionFunction.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/AsyncWriteChannelOptions.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright 2014 Google Inc. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/ResilientOperation.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/PropertyUtil.java
* /opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/RetryBoundedBackOff.java
*
/opt/cola/permits/176126637_1695325572.5580323/0/util-1-4-5-sources-3-jar/com/google/cloud/hadoop/util/RetryDeterminer.java

1.163 config 1.3.0

1.164 asm-all-repackaged 2.2.0

1.164.1 Available under license :

No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*

```
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Creates a new {@link GeneratorAdapter}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the
   * {@link #GeneratorAdapter(int, MethodVisitor, int, String, String)}
   * version.
   *
   * @param mv
   *            the
 method visitor to which this adapter delegates calls.
   * @param access
   *            the method's access flags (see {@link Opcodes}).
   * @param name
   *            the method's name.
   * @param desc
   *            the method's descriptor (see {@link Type Type}).
   */
```

Found in path(s):

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/GeneratorAdapter.java

No license file was found, but licenses were detected in source scan.

#All rights reserved.

```
#Redistribution and use in source and binary forms, with or without
#modification, are permitted provided that the following conditions
#are met:
#1. Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
#2. Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the distribution.
#3. Neither the name of the copyright holders nor the names of its
#   this software without specific prior written permission.
```

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/shrink-frames.properties
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/shrink.properties
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/shrink-writer.properties
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/shrink-resize.properties
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/shrink-signatures.properties
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/shrink-annotations.properties

No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2013 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
```

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/TypePath.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/TypeReference.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/pom.xml
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/Textifiable.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/ASMifiable.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/ASMContentHandler.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/SAXAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
*
 /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/SAXCodeAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-

jar/org/objectweb/asm/xml/SAXClassAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/Processor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/SAXFieldAdapter.java
No license file was found, but licenses were detected in source scan.

<!--
* ASM XML Adapter
* Copyright (c) 2004-2011, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
-->

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/package.html
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*

```
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/Method.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/ASMifier.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Frame.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/SimpleRemapper.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/Interpreter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/BasicValue.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/AnnotationConstantsCollector.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Type.java

*

/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/Value.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/TraceFieldVisitor.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/MethodInsnNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/FrameNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/AnnotationWriter.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/JarOptimizer.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/FieldNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/MultiANewArrayInsnNode.java

*

/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Item.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/MethodNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Handle.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/AnnotationNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/Analyzer.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/SourceInterpreter.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/Printer.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/FieldWriter.java

*

/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/ClassWriter.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/TypeInsnNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/LineNumberNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/TypeAnnotationNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/LdcInsnNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/ClassVisitor.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-

jar/org/objectweb/asm/tree/InsnNode.java
*
/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/ConstantPool.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/CheckAnnotationAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/IntInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/TraceClassVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/ClassConstantsCollector.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/TableSwitchInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/RemappingClassAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/Constant.java
*
/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/TraceSignatureVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/CodeSizeEvaluator.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/ClassOptimizer.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/InstructionAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/SimpleVerifier.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/FieldConstantsCollector.java
*
/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/JumpInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/CheckClassAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/TableSwitchGenerator.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/AnnotationVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/SmallSet.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/TryCatchBlockNode.java

*

/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/InnerClassNode.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/LabelNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/VarInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/AbstractInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/AdviceAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/IincInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Context.java

*

/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/SourceValue.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/Frame.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/InsnList.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/Remapper.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/Shrinker.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/LookupSwitchInsnNode.java

*

/opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Edge.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/CheckFieldAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/TryCatchBlockSorter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/RemappingSignatureAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/signature/SignatureVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-

jar/org/objectweb/asm/commons/RemappingMethodAdapter.java
*
 /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/TraceMethodVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/StaticInitMerger.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/AnalyzerException.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/MethodOptimizer.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Opcodes.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/SerialVersionUIDAdder.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Attribute.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/ByteVector.java
*
 /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/LocalVariablesSorter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/ParameterNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/LocalVariableNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/RemappingFieldAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/InvokeDynamicInsnNode.java
*
 /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/NameMapping.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/RemappingAnnotationAdapter.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/BasicVerifier.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/Subroutine.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/FieldInsnNode.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/CheckMethodAdapter.java

* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/optimizer/MethodConstantsCollector.java

No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Constructs a new {@link LocalVariableAnnotationNode}. <i>Subclasses must
   * not use this constructor</i>. Instead, they must use the
   * {@link #LocalVariableAnnotationNode(int, TypePath, LabelNode[], LabelNode[], int[], String)}
   * version.
   *


   * @param typeRef
   *            a reference to the annotated type. See {@link TypeReference}.
   * @param typePath
   *            the path to the annotated type argument, wildcard bound, array
   *            element type, or static inner type within 'typeRef'. May be
   *            <tt>null</tt> if the annotation targets 'typeRef' as a whole.
   * @param start
   *          the fist instructions corresponding to the continuous ranges
   *          that make the scope of this local variable (inclusive).
   * @param end
   *          the last instructions corresponding to the continuous ranges
   *          that make the scope of this local variable (exclusive). This
   *          array must have the same size as the 'start' array.
   * @param index
   *          the local variable's index in each range. This array must have
   *          the same size as the 'start' array.
   * @param desc
   *          the class descriptor of
 the annotation class.
   */
```

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/LocalVariableAnnotationNode.java
No license file was found, but licenses were detected in source scan.

```
*/
/**
   * Creates a new JSRInliner. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the
   * {@link #JSRInlinerAdapter(int, MethodVisitor, int, String, String, String, String[])}
   * version.
   *
   * @param mv
   *
  the <code>MethodVisitor</code> to send the resulting inlined
   *            method code to (use <code>null</code> for none).
   * @param access
   *            the method's access flags (see {@link Opcodes}). This
   *            parameter also indicates if the method is synthetic and/or
   *            deprecated.
   * @param name
   *            the method's name.
   * @param desc
   *            the method's descriptor (see {@link Type}).
   * @param signature
   *            the method's signature. May be <tt>null</tt>.
   * @param exceptions
   *            the internal names of the method's exception classes (see
   *            {@link Type#getInternalName() getInternalName}). May be
   *            <tt>null</tt>.
   */
```

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
No license file was found, but licenses were detected in source scan.

*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
* Creates a new {@link AnalyzerAdapter}. <i>Subclasses must not use this
* constructor</i>. Instead, they must use the
* {@link #AnalyzerAdapter(int, String, int, String, String, MethodVisitor)}
* version.
*
* @param owner
*
the owner's class name.
* @param access
*   the method's access flags (see {@link Opcodes}).
* @param name
*   the method's name.
* @param desc
*   the method's descriptor (see {@link Type Type}).
* @param mv
*   the method visitor to which this adapter delegates calls. May
*   be <tt>null</tt>.
*/

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/AnalyzerAdapter.java
No license file was found, but licenses were detected in source scan.

<!--
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/attrs/package.html
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/package.html
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/analysis/package.html
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/signature/package.html
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/commons/package.html
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/package.html
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/package.html

No license file was found, but licenses were detected in source scan.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/xml/asm-xml.dtd
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
```

```
/**
   * Constructs a new {@link Textifier}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the {@link #Textifier(int)}
   * version.
   */
```

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/util/Textifier.java
No license file was found, but licenses were detected in source scan.

```
/**
   * Constructs a new {@link ClassNode}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the {@link #ClassNode(int)}
   * version.
   */
```

Found in path(s):
* /opt/cola/permits/176465764_1695326960.9626064/0/asm-all-repackaged-2-2-0-sources-jar/org/objectweb/asm/tree/ClassNode.java

1.165 cglib 2.2.0

1.165.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/Factory.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/UndeclaredThrowableException.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/MethodInterceptor.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/BeanMap.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/MethodInterceptorGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/ImmutableBean.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/KeyFactory.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/VisibilityPredicate.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassFilterTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/BeanMapEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/CallbackGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/DebuggingClassWriter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/DispatcherGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/MethodProxy.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/FastClass.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/EmitUtils.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/FixedValueGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/TypeUtils.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/DefaultNamingPolicy.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/LazyLoaderGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/Mixin.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/Proxy.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ReflectUtils.java
*

/opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/CollectionUtils.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassTransformerChain.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/MethodWrapper.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/BulkBeanEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/MulticastDelegate.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/AbstractClassLoader.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/NoOpGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/InvocationHandlerGenerator.java
*
/opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/AbstractClassGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/MethodDelegate.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/MixinEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/FastClassEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/AddStaticInitTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/FastMethod.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/BeanCopier.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/CodeEmitter.java
*
/opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/CallbackFilter.java
No license file was found, but licenses were detected in source scan.

<!--

obtain a copy of the License at
https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
or packager/legal/LICENSE.txt. See the License for the specific
language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each
file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath"
exception as provided by Oracle in the GPL Version 2 section of the License
file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields
enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license." If you don't indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above. However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if
the new code is made subject to such option by the copyright
holder.

-->

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/pom.xml
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2003 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/InterceptFieldTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/FieldVisitorTee.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/FastConstructor.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/util/StringSwitcher.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/MixinBeanEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/AbstractProcessTask.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/ProxyRefDispatcher.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Block.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Customizer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/Dispatcher.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassTransformer.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/MethodFilter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Predicate.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassTransformerTee.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/MethodFilterTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/CodeGenerationException.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ProcessSwitchCallback.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassReaderGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/DuplicatesPredicate.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/BulkBeanException.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Constants.java

* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/FixedKeySet.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/UndeclaredThrowableTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassVisitorTee.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/AnnotationVisitorTee.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/NoOp.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/InterceptFieldEnabled.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/TinyBitSet.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/NamingPolicy.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Converter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/LazyLoader.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/FieldProvider.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ObjectSwitchCallback.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/InvocationHandler.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/ConstructorDelegate.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/AddPropertyTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ProcessArrayCallback.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Local.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/beans/BeanGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassTransformerFactory.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ClassGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/InterceptFieldCallback.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/DefaultGeneratorStrategy.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/UndeclaredThrowableStrategy.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ClassEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-

jar/net/sf/cglib/beans/BulkBean.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/FieldProviderTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ClassNameReader.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/InterceptFieldFilter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassFilter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/TransformingClassGenerator.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/AbstractClassTransformer.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/AccessFieldTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/ClassEmitterTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Signature.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/TransformingClassLoader.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/AddDelegateTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/Transformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/Callback.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/MethodVisitorTee.java
*
 /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/util/ParallelSorter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/reflect/FastMember.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/GeneratorStrategy.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ClassesKey.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/util/ParallelSorterEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/FixedValue.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/util/SorterTemplate.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2002,2003,2004 The Apache Software Foundation

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/Enhancer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2003,2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/AbstractTransformTask.java
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2005 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:

* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/LocalVariablesSorter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2004 The Apache Software Foundation
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/MethodInfo.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-

jar/net/sf/cglib/proxy/CallbackHelper.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/AbstractClassFilterTransformer.java
*
/opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/ClassInfo.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/InterfaceMaker.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/MethodInfoTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/AbstractInterceptFieldCallback.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/transform/impl/AddInitTransformer.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/MixinEverythingEmitter.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/proxy/CallbackInfo.java
* /opt/cola/permits/176470075_1695326973.8215299/0/cglib-2-2-0-b21-sources-jar/net/sf/cglib/core/RejectModifierPredicate.java

1.166 hk2-api 2.2.0

1.166.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/pom.xml
No license file was found, but licenses were detected in source scan.

* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/PostConstruct.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/jvnet/hk2/annotations/InhabitantAnnotation.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/jvnet/hk2/annotations/package-info.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/jvnet/hk2/annotations/Service.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/jvnet/hk2/annotations/Contract.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2007-2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.

*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made
 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/DescriptorBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/jvnet/hk2/annotations/Optional.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/BuilderHelper.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* http://glassfish.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.

*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception as provided
by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject to
such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/BindingBuilderFactory.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/ServiceBindingBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/ScopedNamedBindingBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/AbstractBindingBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/BindingBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/NamedBindingBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/ScopedBindingBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/ServiceLocatorFactoryImpl.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/DescriptorType.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Operation.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-

jar/org/glassfish/hk2/api/UseProxy.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/StarFilter.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/HK2LoaderImpl.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Filter.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/InjecteeImpl.java
*
/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/HK2RuntimeException.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/InstanceLifecycleListener.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Factory.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/AnnotationLiteral.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/SingleCache.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Proxiable.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/DescriptorBuilderImpl.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Validator.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/ActiveDescriptorBuilderImpl.java
*
/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/HK2Exception.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/PerLookup.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ActiveDescriptor.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/DescriptorVisibility.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Injectee.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Descriptor.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ErrorService.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Metadata.java
*
/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/SpecificFilterImpl.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-

jar/org/glassfish/hk2/api/ErrorInformation.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/FactoryDescriptors.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/InjectionResolver.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/ActiveDescriptorBuilder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ValidationInformation.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Self.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/IndexedFilter.java
*
/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/HK2Loader.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/InstanceLifecycleEventType.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/JustInTimeInjectionResolver.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/AbstractActiveDescriptor.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/extension/ServiceLocatorGenerator.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/MultiException.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/PerThreadContext.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/ServiceLocatorUtilities.java
*
/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/InstanceLifecycleEvent.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/ConstantActiveDescriptor.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Visibility.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/DynamicConfigurationService.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ProxyCtl.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ServiceLocatorFactory.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Unqualified.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/Binder.java
*

/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ServiceLocator.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/UnsatisfiedDependencyException.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Unproxiable.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/DynamicConfiguration.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/AliasDescriptor.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Context.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/FactoryDescriptorsImpl.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/DescriptorImpl.java

*

/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/NamedImpl.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/jvnet/hk2/annotations/ContractsProvided.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ValidationService.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ServiceLocatorState.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/IndexedFilterImpl.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/IterableProvider.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/PerThread.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ServiceHandle.java

*

/opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ErrorType.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Rank.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/binding/AbstractBinder.java
No license file was found, but licenses were detected in source scan.

```
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt.  See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license."  If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above.  However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/
```

Found in path(s):

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/PreDestroy.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/TypeLiteral.java

No license file was found, but licenses were detected in source scan.

```
/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License").  You
* may not use this file except in compliance with the License.  You can
```

Found in path(s):
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/DuplicatePostProcessor.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/DescriptorFileFinder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Immediate.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/Populator.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/PopulatorPostProcessor.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/ImmediateContext.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/internal/ImmediateHelper.java

* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/ImmediateErrorHandler.java
*
 /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/ClasspathDescriptorFileFinder.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/DynamicConfigurationListener.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/InterceptionService.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ProxyForSameScope.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/utilities/RethrowErrorService.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ServiceLocatorListener.java
* /opt/cola/permits/176474360_1695327604.0698712/0/hk2-api-2-2-0-sources-jar/org/glassfish/hk2/api/ClassAnalyzer.java

1.167 hk2-locator 2.2.0

1.167.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/pom.xml
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/AutoActiveDescriptor.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/NarrowResults.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ConstantActiveDescriptor.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/InjecteeImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/InstanceLifecycleEventImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/DelegatingClassLoader.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ClazzCreator.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/Utilities.java
*
/opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/CacheKey.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/MethodInterceptorImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/SystemDescriptor.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/IterableProviderImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ThreeThirtyResolver.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ImmediateResults.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-

jar/org/jvnet/hk2/internal/IndexedListData.java
*
 /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ValidationInformationImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/external/generator/ServiceLocatorGeneratorImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/SingletonContext.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ServiceHandleImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/Constants.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ServiceHandleComparator.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/DescriptorComparator.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/DynamicConfigurationImpl.java
*
 /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/Creator.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/TypeChecker.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/PerLookupContext.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/Collector.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ErrorInformationImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/DynamicConfigurationServiceImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ErrorResults.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/FactoryCreator.java
*
 /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ServiceLocatorImpl.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/DefaultClassAnalyzer.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/MethodInterceptorHandler.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/PopulatorImpl.java
* /opt/cola/permits/176474458_1695326334.5873315/0/hk2-locator-2-2-0-sources-2-jar/org/jvnet/hk2/internal/ConstructorInterceptorHandler.java

1.168 hk2-utils 2.2.0

1.168.1 Available under license :

No license file was found, but licenses were detected in source scan.

<!--

DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Copyright (c) 2010-2011 Oracle and/or its affiliates. All rights reserved.

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can obtain a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html or packager/legal/LICENSE.txt. See the License for the specific language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each file and include the License file at packager/legal/LICENSE.txt.

GPL Classpath Exception:
Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the GPL Version 2 section of the License file that accompanied this code.

Modifications:
If applicable, add the following below the License Header, with the fields enclosed by brackets [] replaced by your own identifying information:
"Portions Copyright [year] [name of copyright owner]"

Contributor(s):
If you wish your version of this file to be governed by only the CDDL or only the GPL Version 2, indicate your decision by adding "[Contributor] elects to include this software in this distribution under the [CDDL or GPL Version 2] license." If you don't indicate a single choice of license, a recipient has the option to distribute your version of this file under either the CDDL, the GPL Version 2 or to extend the choice of license to its licensees as provided above. However, if you add GPL Version 2 code and therefore, elected the GPL Version 2 license, then the option applies only if
the new code is made subject to such option by the copyright holder.

-->

Found in path(s):
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/pom.xml

No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject
 to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/reflection/ParameterizedTypeImpl.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-

jar/org/glassfish/hk2/utilities/cache/LRUCache.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/reflection/Logger.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/CacheEntry.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/reflection/Pretty.java
No license file was found, but licenses were detected in source scan.

 subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/jvnet/hk2/component/MultiMap.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2012-2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
 as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made

subject to such option by the copyright
* holder.
*/

Found in path(s):
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/reflection/ReflectionHelper.java
No license file was found, but licenses were detected in source scan.

/*
* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
*
* Copyright (c) 2013 Oracle and/or its affiliates. All rights reserved.
*
* The contents of this file are subject to the terms of either the GNU
* General Public License Version 2 only ("GPL") or the Common Development
* and Distribution License("CDDL") (collectively, the "License"). You
* may not use this file except in compliance with the License. You can
* obtain a copy of the License at
* https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
* or packager/legal/LICENSE.txt. See the License for the specific
* language governing permissions and limitations under the License.
*
* When distributing the software, include this License Header Notice in each
* file and include the License file at packager/legal/LICENSE.txt.
*
* GPL Classpath Exception:
* Oracle designates this particular file as subject to the "Classpath"
* exception
as provided by Oracle in the GPL Version 2 section of the License
* file that accompanied this code.
*
* Modifications:
* If applicable, add the following below the License Header, with the fields
* enclosed by brackets [] replaced by your own identifying information:
* "Portions Copyright [year] [name of copyright owner]"
*
* Contributor(s):
* If you wish your version of this file to be governed by only the CDDL or
* only the GPL Version 2, indicate your decision by adding "[Contributor]
* elects to include this software in this distribution under the [CDDL or GPL
* Version 2] license." If you don't indicate a single choice of license, a
* recipient has the option to distribute your version of this file under
* either the CDDL, the GPL Version 2 or to extend the choice of license to
* its licensees as provided above. However, if you add GPL Version 2 code
* and therefore, elected the GPL Version 2 license, then the option applies
* only if the new code is made subject

Found in path(s):
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/Computable.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/LRUHybridCache.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/CacheKeyFilter.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/HybridCacheEntry.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/Cache.java
* /opt/cola/permits/176474556_1695326999.295102/0/hk2-utils-2-2-0-sources-jar/org/glassfish/hk2/utilities/cache/internal/LRUCacheCheapRead.java

1.169 xstream 1.3.1

1.169.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/EncodedByteArrayConverter.java

No license file was found, but licenses were detected in source scan.

*
* Created on 07. April 2006 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/CharsetConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 09. October 2008 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/AbstractAttributeAliasingMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/SystemAttributeAliasingMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 03. May 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/XStream11XmlFriendlyMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/AbstractXmlFriendlyMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/ExtendedHierarchicalStreamWriterHelper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/ExtendedHierarchicalStreamWriter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/ThreadSafePropertyEditor.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/HarmonyReflectionProvider.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 12. April 2006 by Joerg Schaible
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XomDriver.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 02. March 2006 by Joerg Schaible
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/AbstractReflectionConverter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 08.12.2007 by Joerg Schaible
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/LookAndFeelConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 24. August 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/ExternalizableConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 18. November 2008 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/AbstractFilePersistenceStrategy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 22. April 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XStream11XmlFriendlyReplacer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 07. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/PureJavaReflectionProvider.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/Dom4JXmlWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/path/PathTracker.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/QuickWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XppDomWriter.java
*
/opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/Dom4JDriver.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/Dom4JReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/path/PathTrackingWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/DomReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/ReflectionConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XppDomReader.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/SelfStreamingInstanceChecker.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/AbstractXmlDriver.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/package.html
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/package.html
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 01. October 2004 by James Strachan
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/QNameMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 23. February 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/PropertiesConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 15. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-

jar/com/thoughtworks/xstream/core/TreeMarshaller.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 14. August 2004 by Joe Walnes
*/
/**
* A SAX {@link org.xml.sax.XMLReader parser} that acts as an XStream
* {@link com.thoughtworks.xstream.io.HierarchicalStreamWriter} to enable direct generation of a
* SAX event flow from the XStream serialization of a list of list of Java objects. <p/> As a
* custom SAX parser, this class ignores the arguments of the two standard parse methods ({@link
#parse(java.lang.String)}
* and {@link #parse(org.xml.sax.InputSource)}) but relies on a proprietary SAX property
* {@link #SOURCE_OBJECT_LIST_PROPERTY} to define the list of objects to serialize.
* </p>
* <p/> Configuration
 of this SAX parser is achieved through the standard
* {@link #setProperty SAX property mechanism}. While specific setter methods require direct
* access to the parser instance, SAX properties support configuration settings to be propagated
* through a chain of {@link org.xml.sax.XMLFilter filters} down to the underlying parser
* object.
* </p>
* <p/> This mechanism shall be used to configure the
* {@link #SOURCE_OBJECT_LIST_PROPERTY objects to be serialized} as well as the
* {@link #CONFIGURED_XSTREAM_PROPERTY XStream facade}.
* </p>
*
* @author Laurent Bihanic
*/
/**
   * Sets the value of a property.
   * <p/>
   * The property name is any fully-qualified URI.  It is
   * possible for an XMLReader to recognize a property name but
   * to be unable to set its value.</p>
   * <p/>
   * XMLReaders are not required to recognize setting any
   * specific property names, though a core set is provided with
   * SAX2.</p>
   * <p/>
```

```
    * Some property
  values may be immutable or mutable only
    * in specific contexts, such as before, during, or after
    * a parse.</p>
    * <p/>
    * This method is also the standard mechanism for setting
    * extended handlers.</p>
    * <p/>
    * <strong>Note</strong>: This implementation only supports two
    * (proprietary) properties: {@link #CONFIGURED_XSTREAM_PROPERTY}
    * and {@link #SOURCE_OBJECT_LIST_PROPERTY}.</p>
    *
    * @param name  the property name, which is a fully-qualified URI.
    * @param value the requested value for the property.
    * @throws SAXNotRecognizedException when the XMLReader does not
    *                                   recognize the property name.
    * @throws SAXNotSupportedException  when the XMLReader recognizes
    *                                   the property name but cannot set
    *                                   the requested value.
    * @see #getProperty
    */
 /**
    * Parses an XML document from a system identifier
  (URI).
    * <p/>
    * This method is a shortcut for the common case of reading a
    * document from a system identifier.  It is the exact
    * equivalent of the following:</p>
    * <blockquote>
    * <pre>
    *    parse(new InputSource(systemId));
    *  </pre>
    * </blockquote>
    * <p/>
    * If the system identifier is a URL, it must be fully resolved
    * by the application before it is passed to the parser.</p>
    * <p/>
    * <strong>Note</strong>: As a custom SAX parser, this class
    * ignores the <code>systemId</code> argument of this method
    * and relies on the proprietary SAX property
    * {@link #SOURCE_OBJECT_LIST_PROPERTY}) to define the list of
    * objects to serialize.</p>
    *
    * @param systemId the system identifier (URI).
    * @throws SAXException Any SAX exception, possibly wrapping
    *                      another exception.
    * @see #parse(org.xml.sax.InputSource)
    */
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/SaxWriter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/FieldDictionary.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/enums/EnumSingleValueConverter.java
No license file was found, but licenses were detected in source scan.

* Created on 02. September 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/path/Path.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/FastStack.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 30. May 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/StackTraceElementFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003, 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 29. September 2003 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/StringBufferConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.

```
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 29. May 2004 by Joe Walnes
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/StackTraceElementConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/ThrowableConverter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 16. March 2004 by Joe Walnes
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/SequenceGenerator.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 06. November 2007 by Joerg Schaible
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/LocalConversionMapper.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 24. April 2005 by Joe Walnes
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/AbstractDocumentReader.java

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/AbstractPullReader.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 03. October 2005 by Joerg Schaible
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/ISO8601SqlTimestampConverter.java

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/ISO8601GregorianCalendarConverter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 26.09.2007 by Joerg Schaible
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/AbstractTreeMarshallingStrategy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 02. March 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/AnnotationProvider.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/AnnotationReflectionConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 01. February 2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/AbstractAttributedCharacterIteratorAttributeConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*

* Created on 04. January 2008 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/UUIDConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/StringBuilderConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 30. March 2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/json/JettisonMappedXmlDriver.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 16. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/ReferenceByIdMarshallingStrategy.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/TreeMarshallingStrategy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamAsAttribute.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/PersistenceStrategy.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/FilePersistenceStrategy.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/Dom4JWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/HierarchicalStreamDriver.java

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/HierarchicalStreamWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/xppdom/Xpp3DomBuilder.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/HierarchicalStreamReader.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 24. July 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/SqlDateConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/CurrencyConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/SqlTimeConverter.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 03. April 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/ReferenceByXPathUnmarshaller.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/BigIntegerConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/BigDecimalConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/DataHolder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-

jar/com/thoughtworks/xstream/persistence/XmlArrayList.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/InitializationException.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/XStreamException.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/ReaderWrapper.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/DefaultImplementationsMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/ArrayMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/CachingMapper.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/ByteConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/SerializationMethodInvoker.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/UnmarshallingContext.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/ErrorWriter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/JavaClassConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/JavaMethodConverter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 04. June 2006 by Joe Walnes
*/
```

Found in path(s):

```
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/copy/HierarchicalStreamCopier.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 08. May 2004 by Joe Walnes
*/
```

Found in path(s):

```
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/TreeMapConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/TreeSetConverter.java
```

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 04. October 2004 by Joe Walnes
```

*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/MapBackedDataHolder.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/javabean/PropertyDictionary.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/javabean/JavaBeanConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/javabean/BeanProperty.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/javabean/BeanProvider.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/Pool.java
No license file was found, but licenses were detected in source scan.

* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 31. July 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/RegexPatternConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003, 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 01. October 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/SqlTimestampConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 29. September 2004 by James Strachan
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/StaxWriter.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 29. September 2004 by James Strachan
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/StaxReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/StaxDriver.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 24. July 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/GregorianCalendarConverter.java
No license file was found, but licenses were detected in source scan.

<!--
Copyright (C) 2004 Joe Walnes.
Copyright (C) 2006, 2007 XStream committers.
All rights reserved.

The software in this package is published under the terms of the BSD
style license a copy of which has been included with this distribution in
the LICENSE.txt file.

Created on 10. May 2004 by Joe Walnes

-->

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/package.html
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/alias/ClassMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/ConverterLookup.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/StringConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/AbstractCollectionConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/ConversionException.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/MapConverter.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 25. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/DynamicProxyConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 30. March 2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/TypedNull.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/DependencyInjectionFactory.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003, 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 03. October 2003 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/NullConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 08. July 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/FontConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 16. February 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/ImplicitCollectionMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.

*
* Created on 21. December 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/SerializableConverter.java
No license file was found, but licenses were detected in source scan.

*
* Created on 16. September 2005 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamImplicitCollection.java
No license file was found, but licenses were detected in source scan.

*
* Created on 25. March 2006 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/TextAttributeConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/ReflectionProviderWrapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/XStreamer.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/HierarchicalStreams.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/xppdom/Xpp3Dom.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/ObjectAccessException.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-

jar/com/thoughtworks/xstream/converters/collections/BitSetConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/MarshallingStrategy.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/FastField.java
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/ClassAliasingMapper.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 09. May 2004 by Joe Walnes
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/JVM.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 22. November 2004 by Mauro Talevi
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/ISO8601DateConverter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
```

* Created on 20. February 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/AttributeMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 17.05.2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/NativeFieldKeySorter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 07. March 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/ClassLoaderReference.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.

*
* Created on 18. October 2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/DocumentReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/DocumentWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/AbstractDocumentWriter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 26. September 2003 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/CannotResolveClassException.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/alias/CannotResolveClassException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 13. January 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/FileConverter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 16. November 2004 by Joe Walnes
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/CompositeClassLoader.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 01. December 2006 by Joerg Schaible
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamImplicit.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2003, 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 26. September 2003 by Joe Walnes
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/XStream.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/AnnotationMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/AnnotationConfiguration.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/Fields.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamOmitField.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/Sun14ReflectionProvider.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/DateConverter.java
No license file was found, but licenses were detected in source scan.

* the LICENSE.txt file.
*
* Created on 08. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XppReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XppDriver.java
No license file was found, but licenses were detected in source scan.

<!--
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 10. April 2004 by Joe Walnes
-->

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/path/package.html
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 13. April 2006 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/CGLIBEnhancedConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.

* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 09. April 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/FieldAliasingMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 01. January 2008 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/ConverterRegistry.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 21.09.2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/DurationConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.

* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 07. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/CompactWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/DomDriver.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XppDomDriver.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/PrettyPrintWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/DefaultConverterLookup.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 19.09.2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/XStream12FieldKeySorter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*

* Created on 06. February 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/PrioritizedList.java
No license file was found, but licenses were detected in source scan.

* Created on 14. September 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/BaseException.java
No license file was found, but licenses were detected in source scan.

* Created on 11. August 2005 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamAlias.java
No license file was found, but licenses were detected in source scan.

* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 20. March 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/EnumMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 06. April 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/enums/EnumSetConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/enums/EnumMapConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 22. June 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/json/JsonHierarchicalStreamDriver.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/json/JsonHierarchicalStreamWriter.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 15. February 2006 by Mauro Talevi
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/AbstractSingleValueConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/SingleValueConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/ConverterMatcher.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 06. March 2004 by Joe Walnes
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/CharArrayConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/MarshallingContext.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
```

* the LICENSE.txt file.
*
* Created on 06. May 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/ThreadSafeSimpleDateFormat.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/ThreadSafeSimpleDateFormat.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 11. August 2005 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/Annotations.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamContainedType.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 20.09.2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/PropertyEditorCapableConverter.java
No license file was found, but licenses were detected in source scan.

/*

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/SubjectConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/ReferenceByXPathMarshallingStrategy.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/path/PathTrackingReader.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-

jar/com/thoughtworks/xstream/io/json/JsonWriter.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 08. April 2006 by Joerg Schaible
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/CGLIBMapper.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 25. March 2004 by Joe Walnes
*/
```

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/URLConverter.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 11. October 2006 by Joerg Schaible
*/
```

Found in path(s):

* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/Primitives.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 10. November 2008 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/PackageAliasingMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 13. November 2008 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamInclude.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 10. April 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/WriterWrapper.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/enums/EnumConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/ToStringConverter.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/XmlMap.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/StreamStrategy.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/FileStreamStrategy.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/CustomObjectInputStream.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/CustomObjectOutputStream.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/OuterClassMapper.java
No license file was found, but licenses were detected in source scan.

/*

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/TraxSource.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/JDomReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/JDomDriver.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XomWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/JDomWriter.java
No license file was found, but licenses were detected in source scan.

* Created on 28. June 2006 by Guilherme Silveira
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/persistence/XmlSet.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003, 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 26. September 2003 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/IntConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/ShortConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/CharConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/BooleanConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/FloatConverter.java
*
/opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/LongConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/AbstractBasicConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/Converter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/basic/DoubleConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in

* the LICENSE.txt file.
*
* Created on 20. February 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/SingleValueConverterWrapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 22. January 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/MapperWrapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/DefaultMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/Mapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/XmlFriendlyMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 15. March 2007 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/AbstractReferenceMarshaller.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-

jar/com/thoughtworks/xstream/core/AbstractReferenceUnmarshaller.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/LocaleConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/SortableFieldKeySorter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/FieldKey.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/FieldKeySorter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/ImmutableFieldKeySorter.java
No license file was found, but licenses were detected in source scan.

* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 04. June 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/AbstractXmlReader.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 06. August 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/Base64Encoder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 03. April 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/ReferenceByXPathMarshaller.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2003, 2004, 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.

* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 01. October 2003 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/extended/ColorConverter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/CollectionConverter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 27. Februar 2004 by Jason van Zyl
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/alias/NameMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 24. April 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/AttributeNameIterator.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 06. February 2005 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/OrderRetainingMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 07. March 2004 by Joe Walnes
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/reflection/ReflectionProvider.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/StreamException.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 22. January 2005 by Joe Walnes

*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/ImmutableTypesMapper.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/DynamicProxyMapper.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 04. June 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/AbstractXmlWriter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 15. March 2006 by Joerg Schaible
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/StatefulWriter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XmlFriendlyReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XmlFriendlyWriter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/converters/collections/ArrayConverter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/binary/BinaryStreamWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/binary/BinaryStreamReader.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-

jar/com/thoughtworks/xstream/io/binary/ReaderDepthState.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/binary/Token.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 13. September 2007 by Joerg Schaible.
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/XmlHeaderAwareReader.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 17. April 2006 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XmlFriendlyReplacer.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004, 2005, 2006 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 15. March 2004 by Joe Walnes

*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/ReferenceByIdUnmarshaller.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/TreeUnmarshaller.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/ReferenceByIdMarshaller.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2008 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 17.04.2008 by Joerg Schaible.
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/json/JettisonStaxWriter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2005 Joe Walnes.
* Copyright (C) 2006, 2007 XStream Committers.
* All rights reserved.
*
* The software in this package is published under the terms of the BSD
* style license a copy of which has been included with this distribution in
* the LICENSE.txt file.
*
* Created on 16. September 2005 by Mauro Talevi
*/

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/annotations/XStreamConverters.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Joe Walnes.
* Copyright (C) 2006, 2007, 2008 XStream Committers.
* All rights reserved.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/core/util/ObjectIdDictionary.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/mapper/AttributeAliasingMapper.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/DomWriter.java
* /opt/cola/permits/176485837_1695327610.0925384/0/xstream-1-3-1-sources-jar/com/thoughtworks/xstream/io/xml/XomReader.java

1.170 httpcore-nio 4.4.6

1.170.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

Apache HttpCore NIO
Copyright 2005-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.171 react-bootstrap 0.32.1

1.171.1 Available under license :

1.172 aspectj-tools 1.8.13

1.173 gson 2.8.2

1.173.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/annotations/JsonAdapter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/$Gson$Types.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonStreamParser.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/FieldAttributes.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonParser.java
*
 /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/LongSerializationPolicy.java
No license file was found, but licenses were detected in source scan.

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/stream/MalformedJsonException.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/stream/JsonReader.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/stream/JsonWriter.java
*
/opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/stream/JsonToken.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/stream/JsonScope.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
*

/opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/JsonTreeWriter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/TypeAdapter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/TypeAdapterFactory.java
*
/opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/UnsafeAllocator.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/SqlDateTypeAdapter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/TreeTypeAdapter.java
*
/opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/TimeTypeAdapter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/LazilyParsedNumber.java

* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/ConstructorConstructor.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/bind/DateTypeAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/Streams.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonObject.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-

jar/com/google/gson/FieldNamingPolicy.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/GsonBuilder.java
*
 /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/JsonSerializationContext.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonNull.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/annotations/Since.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/ExclusionStrategy.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/reflect/TypeToken.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonPrimitive.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/DefaultDateTypeAdapter.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/JsonDeserializationContext.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/InstanceCreator.java
*
 /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/JsonIOException.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/JsonParseException.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/internal/ObjectConstructor.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/annotations/SerializedName.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/annotations/Until.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonElement.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/JsonDeserializer.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonSerializer.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/internal/Primitives.java
*
 /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/internal/$Gson$Preconditions.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/JsonArray.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/internal/Excluder.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/FieldNamingStrategy.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/Gson.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-
jar/com/google/gson/annotations/Expose.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/LinkedTreeMap.java
* /opt/cola/permits/178318228_1695329011.1264045/0/gson-2-8-2-sources-jar/com/google/gson/internal/LinkedHashTreeMap.java

1.174 commons-beanutils 1.9.2

1.174.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.175 protobuf-java 3.2.0

1.175.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapEntry.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionRegistryFactory.java
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Extension.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionRegistryLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ProtobufArrayList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnknownFieldSet.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractProtobufList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapFieldLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/type.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapEntryLite.java
*

/opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/InvalidProtocolBufferException.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/DoubleArrayList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/FieldSet.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyFieldLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Utf8.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcChannel.java
*
/opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MapField.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormatParseInfoTree.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageLiteOrBuilder.java
*
/opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/WireFormat.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ByteBufferWriter.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnsafeByteOperations.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnsafeUtil.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractParser.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/BlockingService.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageOrBuilder.java

* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/empty.proto
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LongArrayList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UnknownFieldSetLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractMessage.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/SingleFieldBuilderV3.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExperimentalApi.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcUtil.java
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/CodedOutputStream.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/DynamicMessage.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/struct.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/CodedInputStream.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyField.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MutabilityOracle.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Parser.java
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/IntArrayList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/GeneratedMessageLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ProtocolStringList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormatEscaper.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-

jar/com/google/protobuf/TextFormat.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RpcController.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/GeneratedMessageV3.java
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/GeneratedMessage.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/source_context.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/duration.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ByteString.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ExtensionRegistry.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RepeatedFieldBuilder.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/BooleanArrayList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/any.proto
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/NioByteString.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ByteOutput.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ServiceException.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/api.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/RopeByteString.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/wrappers.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Service.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/ProtocolMessageEnum.java
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Descriptors.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/UninitializedMessageException.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/AbstractMessageLite.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyStringArrayList.java

* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/google/protobuf/compiler/plugin.proto
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Internal.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/Message.java
*
 /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/FloatArrayList.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/MessageReflection.java
* /opt/cola/permits/179211514_1695330195.320972/0/protobuf-java-3-2-0-sources-6-jar/com/google/protobuf/LazyStringList.java

1.176 protobuf-java-util 3.2.0

1.176.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
// Copyright 2008 Google Inc.  All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//     * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//     * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):
* /opt/cola/permits/179211619_1695330197.414644/0/protobuf-java-util-3-2-0-sources-jar/com/google/protobuf/util/Timestamps.java
* /opt/cola/permits/179211619_1695330197.414644/0/protobuf-java-util-3-2-0-sources-jar/com/google/protobuf/util/FieldMaskTree.java
*
 /opt/cola/permits/179211619_1695330197.414644/0/protobuf-java-util-3-2-0-sources-jar/com/google/protobuf/util/TimeUtil.java
* /opt/cola/permits/179211619_1695330197.414644/0/protobuf-java-util-3-2-0-sources-jar/com/google/protobuf/util/FieldMaskUtil.java
* /opt/cola/permits/179211619_1695330197.414644/0/protobuf-java-util-3-2-0-sources-jar/com/google/protobuf/util/JsonFormat.java
* /opt/cola/permits/179211619_1695330197.414644/0/protobuf-java-util-3-2-0-sources-jar/com/google/protobuf/util/Durations.java

1.177 request-local 1.0.5

1.177.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/179778209_1695331286.68452/0/request-local-master-zip/request-local-master/index.js
* /opt/cola/permits/179778209_1695331286.68452/0/request-local-master-zip/request-local-master/middleware.js
* /opt/cola/permits/179778209_1695331286.68452/0/request-local-master-zip/request-local-master/lib/request-local.js

1.178 react 16.2.0

1.178.1 Available under license :

1.179 react-dom 16.2.0

1.179.1 Available under license :

```
/**
* Copyright (c) 2013-present, Facebook, Inc.
*
* This source code is licensed under the MIT license found in the
* LICENSE file in the root directory of this source tree.
*
* @flow
*/

'use strict';

module.exports = require('./server.node');
```

1.180 metrics 3.1.0

1.180.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/jsr166e/LongAdder.java?revision=1.14&view=markup
*/
```

Found in path(s):

* /opt/cola/permits/180067738_1695331858.996538/0/metrics-core-3-1-0-sources-jar/com/codahale/metrics/LongAdder.java

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/main/java/util/concurrent/ThreadLocalRandom.java?view=markup
*/
```

Found in path(s):

* /opt/cola/permits/180067738_1695331858.996538/0/metrics-core-3-1-0-sources-jar/com/codahale/metrics/ThreadLocalRandom.java

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/jsr166e/Striped64.java?revision=1.8&view=markup
*/
```

Found in path(s):

* /opt/cola/permits/180067738_1695331858.996538/0/metrics-core-3-1-0-sources-jar/com/codahale/metrics/Striped64.java

1.181 metrics 3.1.5

1.181.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/main/java/util/concurrent/ThreadLocalRandom.java?view=markup
*/
```

Found in path(s):

* /opt/cola/permits/180937470_1695332768.58104/0/fsrc-metrics-core-3-1-5-sources-jar/com/codahale/metrics/ThreadLocalRandom.java

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/jsr166e/LongAdder.java?revision=1.14&view=markup
*/
```

Found in path(s):

* /opt/cola/permits/180937470_1695332768.58104/0/fsrc-metrics-core-3-1-5-sources-jar/com/codahale/metrics/LongAdder.java

No license file was found, but licenses were detected in source scan.

```
/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*
* http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/jsr166e/Striped64.java?revision=1.8&view=markup
*/
```

Found in path(s):

* /opt/cola/permits/180937470_1695332768.58104/0/fsrc-metrics-core-3-1-5-sources-jar/com/codahale/metrics/Striped64.java

1.182 okhttp 3.9.1

1.182.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/platform/Platform.java

No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/NamedRunnable.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-

jar/okhttp3/internal/http/RequestLine.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache/CacheStrategy.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/RouteDatabase.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Route.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/ErrorCode.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Http2.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Response.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Dispatcher.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Handshake.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Request.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache/InternalCache.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/MediaType.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/StatusLine.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Hpack.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/RetryAndFollowUpInterceptor.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/tls/BasicCertificateChainCleaner.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/WebSocketListener.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/tls/BasicTrustRootIndex.java

* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/platform/JdkWithJettyBootPlatform.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/StreamResetException.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/tls/TrustRootIndex.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache2/FileOperator.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/CallServerInterceptor.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache2/Relay.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/platform/AndroidPlatform.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/RealInterceptorChain.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/ws/RealWebSocket.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/ConnectionShutdownException.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/platform/Jdk9Platform.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/UnrepeatableRequestBody.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/WebSocket.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/StreamAllocation.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache/FaultHidingSink.java

* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Cookie.java
*
/opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/CookieJar.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/ConnectionSpecSelector.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/RouteException.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/io/FileSystem.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/HttpUrl.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Authenticator.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Huffman.java
No license file was found, but licenses were detected in source scan.

*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/TlsVersion.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/RealResponseBody.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/ConnectionSpec.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/MultipartBody.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/RealCall.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Protocol.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/HttpMethod.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Header.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/FormBody.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/Version.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/Internal.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/RequestBody.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/ws/WebSocketReader.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache/CacheRequest.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/ws/WebSocketWriter.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Credentials.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Call.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/ResponseBody.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/ws/WebSocketProtocol.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Callback.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Interceptor.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Challenge.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/CipherSuite.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/CertificatePinner.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/PushObserver.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/RouteSelector.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Ping.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Dns.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/OkHttpClient.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Settings.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Cache.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/HttpCodec.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/Util.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Http2Codec.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http1/Http1Codec.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Address.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/HttpHeaders.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/publicsuffix/PublicSuffixDatabase.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/EventListener.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/HttpDate.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Http2Stream.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache/DiskLruCache.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Http2Writer.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Http2Connection.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http2/Http2Reader.java
No license file was found, but licenses were detected in source scan.

/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Headers.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/ConnectInterceptor.java

* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/tls/DistinguishedNameParser.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/cache/CacheInterceptor.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/platform/OptionalMethod.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/Connection.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/connection/RealConnection.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/tls/OkHostnameVerifier.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/http/BridgeInterceptor.java
* /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/ConnectionPool.java
*
 /opt/cola/permits/180966299_1695332775.8581772/0/okhttp-3-9-1-sources-jar/okhttp3/internal/tls/CertificateChainCleaner.java

1.183 jackson-annotations 2.9.0

1.183.1 Available under license :

1.184 log4j-api 2.9.1

1.184.1 Available under license :

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

1.185 httpasyncclient 4.1.3

1.185.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

1.186 json-simple 1.1.1

1.186.1 Available under license :

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.187 scala 2.11.12

1.187.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
*
* A `ClassTag[T]` stores the erased class of a given type `T`, accessible via the `runtimeClass`
* field. This is particularly useful for instantiating `Array`s whose element types are unknown
* at compile time.
*
* `ClassTag`s are a weaker special case of [[scala.reflect.api.TypeTags#TypeTag]]s, in that they
* wrap only the runtime class of a given type, whereas a `TypeTag` contains all static type
* information. That is, `ClassTag`s are constructed from knowing only the top-level class of a
* type, without necessarily knowing all of its argument types. This runtime information is enough
* for runtime `Array` creation.
*
* For example:
* {{{
*   scala> def mkArray[T : ClassTag](elems: T*) = Array[T](elems: _*)
*   mkArray: [T](elems: T*)(implicit evidence$1: scala.reflect.ClassTag[T])Array[T]
*
*   scala> mkArray(42, 13)
*   res0: Array[Int] = Array(42, 13)
*
*   scala> mkArray("Japan","Brazil","Germany")
 *   res1: Array[String] = Array(Japan, Brazil, Germany)
* }}}
*
* See [[scala.reflect.api.TypeTags]] for more examples, or the
* [[http://docs.scala-lang.org/overviews/reflection/typetags-manifests.html Reflection Guide: TypeTags]]
* for more details.
*
*/
```

Found in path(s):

* /opt/cola/permits/183286080_1695335354.2084045/0/scala-library-2-11-12-sources-jar/scala/reflect/ClassTag.scala
No license file was found, but licenses were detected in source scan.

```
/**
  * Latch used to implement waiting on a DefaultPromise's result.
  *
  * Inspired by: http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/main/java/util/concurrent/locks/AbstractQueuedSynchronizer.java
  * Written by Doug Lea with assistance from members of JCP JSR-166
  * Expert Group and released to the public domain, as explained at
  * http://creativecommons.org/publicdomain/zero/1.0/
  */
```

Found in path(s):
* /opt/cola/permits/183286080_1695335354.2084045/0/scala-library-2-11-12-sources-jar/scala/concurrent/impl/Promise.scala

1.188 zookeeper 3.4.10

1.188.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

1.189 cxf-rt-transports-http 3.2.6

1.189.1 Available under license :

Apache CXF
Copyright 2006-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project includes Public Suffix List copied from
https://publicsuffix.org/list/effective_tld_names.dat
licensed under the terms of the Mozilla Public License, v. 2.0
Full license text: <http://mozilla.org/MPL/2.0/>

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.190 cxf-rt-rs-client 3.2.6

1.190.1 Available under license :

Apache CXF
Copyright 2006-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

1.191 annotations 2.0.1

1.191.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially
designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do

these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General

Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of
an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use
the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language.
(Hereinafter, translation is
included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a
software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has

appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is
made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section
6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object

file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work
is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the

copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at

least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified
materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them
and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same
work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or

distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and

all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If
you cannot
distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose
of protecting the
integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is
permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

14.
If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software;
you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school,
if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.192 compiler 0.9.3

1.193 jackson-dataformat-cbor 2.8.10

1.194 jackson-dataformat-smile 2.8.10

1.194.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
#  Copyright 2012 FasterXML.com
#
#  Licensed under the Apache License, Version 2.0 (the "License");
#  you may not use this file except in compliance with the License.
#  You may obtain a copy of the License at
#
#  http://www.apache.org/licenses/LICENSE-2.0
#
#  Unless required by applicable law or agreed to in writing, software
#  distributed under the License is distributed on an "AS IS" BASIS,
#  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#  See the License for the specific language governing permissions and
#  limitations under the License.

com.fasterxml.jackson.dataformat.smile.SmileFactory
```

Found in path(s):

* /opt/cola/permits/184758854_1695340384.3725016/0/jackson-dataformat-smile-2-8-10-sources-jar/META-INF/services/com.fasterxml.jackson.core.JsonFactory

1.195 jackson-dataformat-yaml 2.8.10

1.195.1 Available under license :

This copy of Jackson JSON processor YAML module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.196 jna 4.5.1

1.197 bzip2 1.0.6

1.197.1 Available under license :

This program, "bzip2", the associated library "libbzip2", and all
documentation, are copyright (C) 1996-2010 Julian R Seward. All
rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must
  not claim that you wrote the original software. If you use this
  software in a product, an acknowledgment in the product
  documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must
not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
products derived from this software without specific
prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@bzip.org
bzip2/libbzip2 version 1.0.6 of 6 September 2010

1.198 core-module 1.1.7

1.199 typesafe-config 1.3.0

1.200 commons-csv 1.4

1.200.1 Available under license :

Apache Commons CSV
Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

1.201 animal-sniffer-annotation 1.14

1.202 hdrhistogram 2.1.9

1.203 joda-time 2.9.9

1.203.1 Available under license :

===
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
===
This product includes software developed by
Joda.org (http://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute

copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

1.204 zlib 1.2.11

1.204.1 Available under license :

1.205 javax-inject 1

1.206 jackson 2.8.10

1.206.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.207 checker-compat-qual 2.0.0

1.208 commons-codec 1.6

1.208.1 Available under license :

Apache Commons Codec
Copyright 2002-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

----
src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains
test data from http://aspell.sourceforge.net/test/batch0.tab.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.209 commons-io 2.5

1.209.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.210 guava 25.0

1.211 lz4-java 1.4

1.212 osgi-core 6.0.0

1.212.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.213 slf4j 1.7.25

1.214 snappy-java 1.1.4

1.215 metrics 3.1.2

1.216 snappy 6.1.1

1.216.1 Available under license :

1.217 lucene 7.3.1

1.217.1 Available under license :

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at

http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
```

* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * Neither the name of the copyright holders nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS

Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0

See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
 package is available from
 http://snowball.tartarus.org/

The KStem stemmer in
 analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
 light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The

Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801

Notice
==

Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all

damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification,
copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

1.218 lucene-analyzers-common 7.3.1

1.218.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/.
Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

The following license applies to the Snowball stemmers:

modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * Neither the name of the copyright holders nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. The names "Center for Intelligent
Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
derived from this software without prior written permission. To obtain
permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

The
following license applies to the Morfologik project:

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons

ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in

analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801
Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.219 lucene-backward-codecs 7.3.1

1.219.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
```

* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights

* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * Neither the name of the copyright holders nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. The names "Center for Intelligent
 Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
derived from this software without prior written permission. To obtain
permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The
 following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by

Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java

analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801
Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed
substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of

any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification,
copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.220 lucene-grouping 7.3.1

1.220.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
```

* copyright holder.
*/

The following license applies to the Snowball stemmers:

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

 * Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * Neither the name of the copyright holders nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation

and/or other materials provided with the distribution.

3. The names "Center for Intelligent
 Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
derived from this software without prior written permission. To obtain
permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The
 following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

  * Neither the name of Morfologik nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:

- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in

analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.

See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801
Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for

any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the

foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification,
copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

1.221 lucene-highlighter 7.3.1

1.221.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
* Copyright 2001-2004 Unicode, Inc.
*

* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
 Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright

```
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)

The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

The following license applies to the Snowball stemmers:

The following license applies to the KStemmer:

SUCH DAMAGE.

The
following license applies to the Morfologik project:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the
   distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,

analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)

(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801
Notice
==

1.222 lucene-join 7.3.1

1.222.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

The following license applies to the Snowball stemmers:

The following license applies to the KStemmer:

Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:

- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java

analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
 Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
 licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

 http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801
Notice
==

1.223 lucene-memory 7.3.1

1.223.1 Available under license :

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.

Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

The following license applies to the Snowball stemmers:

Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. The names "Center for Intelligent
Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
derived from this software without prior written permission. To obtain

permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The
 following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
 NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
 This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball

package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)

is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801
Notice
==

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.224 lucene-misc 7.3.1

1.224.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice,
    * this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
    * notice, this list of conditions and the following disclaimer in the
    * documentation and/or other materials provided with the distribution.
    * Neither the name of the copyright holders nor the names of its contributors
    * may be used to endorse or promote products derived from this software
    * without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss

Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,

Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
 light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
 Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
 licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==

Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801
Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed
substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the

program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.225 lucene-queries 7.3.1

1.225.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
```

* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT

NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,

and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801
Notice
===

Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection

with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification,
copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

1.226 lucene-queryparser 7.3.1

1.226.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
 Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
```

* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR

OTHER DEALINGS IN THE SOFTWARE.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE

GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The
following license applies to the Morfologik project:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp

- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)

licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801
Notice
==

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed

substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever
for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification,
copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

1.227 lucene-sandbox 7.3.1

1.227.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the

editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
```

* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

```
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
```

* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

The following license applies to the Morfologik project:

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Morfologik nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Apache Lucene

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and

http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
 package is available from
 http://snowball.tartarus.org/

The KStem stemmer in
 analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
 light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java

analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
 Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
 licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
==

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===

mecab-ipadic-2.7.0-20070801
Notice

===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed
substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to

have agreed and understood, that there is no warranty whatsoever
 for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification,
 copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

1.228 lucene-spatial 7.3.1

1.228.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/.
 Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/
```

The following license applies to the Snowball stemmers:

The following license applies to the KStemmer:

The
following license applies to the Morfologik project:

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE

Apache Lucene

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
 This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.

The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801
Notice
==

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an

action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed
substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever
for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,

modification,
copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

1.229 lucene-spatial-extras 7.3.1

1.229.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are

* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
 will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
 Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products

```
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* Neither the name of the copyright holders nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The

following license applies to the Morfologik project:

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP) http://sgjp.pl/morfeusz/

Apache Lucene

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such

as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,

analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801
Notice
==

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever
 for
the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use,
modification,
 copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.230 lucene-spatial3d 7.3.1

1.230.1 Available under license :

Apache License
Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was

derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/
```

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

```
    * Redistributions of source code must retain the above copyright notice,
   * this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above copyright
   * notice, this list of conditions and the following disclaimer in the
   * documentation and/or other materials provided with the distribution.
   * Neither the name of the copyright holders nor the names of its contributors
   * may be used to endorse or promote products derived from this software
   * without specific prior written permission.
```

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

The
 following license applies to the Morfologik project:

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
 Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
 licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

 http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

 http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===

mecab-ipadic-2.7.0-20070801
Notice
==

officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification,
 copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.231 lucene-suggest 7.3.1

1.231.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim
will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/.
Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller

```
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including
 without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available
 here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others.  All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
* IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
* ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
* IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
* OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*
* Except as contained in this notice, the name of a copyright holder shall not
* be used in advertising or otherwise to promote the sale, use or other
* dealings in this Software without prior written authorization of the
* copyright holder.
*/
```

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
All rights reserved.

Redistribution and use in source and binary forms, with or without

The following license applies to the KStemmer:

The
 following license applies to the Morfologik project:

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons

ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI
PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the
  distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Lucene

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre.
 This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball
 package is available from
 http://snowball.tartarus.org/

The KStem stemmer in

analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish
light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The
Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/)
licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801
Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

1.232 netty 4.1.16

1.233 netty-buffer 4.1.16

1.234 netty-codec 4.1.16

1.235 netty-codec-http 4.1.16

1.236 netty-handler 4.1.16

1.237 netty-resolver 4.1.16

1.238 netty-transport 4.1.16

1.239 t-digest 3.2

1.240 commons-collections 4.1

1.240.1 Available under license :

Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from)
the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication
on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant
of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names
of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law
(such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your
sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

1.241 satori-go-uuid 1.2.0

1.241.1 Available under license :

NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.242 httpcomponents-core 4.4.5

1.242.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==

This project contains annotations in the package org.apache.http.annotation
which are derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Full text: http://creativecommons.org/licenses/by/2.5/legalcode

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED

HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation,

or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2.

Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

to create and reproduce Derivative Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly

by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE

OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Apache HttpCore

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.243 httpcore-nio 4.4.5

1.243.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.244 jopt-simple 5.0.2

1.245 spatial4j 0.7

1.246 google-go-cmp 0.2.0

1.246.1 Available under license :

1.247 go-profile 1.2.1

1.247.1 Available under license :

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.248 annotations 3.0.1

1.249 netty 4.0.27

1.250 golang-snappy 2e65f85

1.250.1 Available under license :

```
# This is the official list of people who can contribute
# (and typically have contributed) code to the Snappy-Go repository.
# The AUTHORS file lists the copyright holders; this file
# lists people.  For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# The submission process automatically checks to make sure
# that people submitting code are listed in this file (by email address).
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
#     http://code.google.com/legal/individual-cla-v1.0.html
#     http://code.google.com/legal/corporate-cla-v1.0.html
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate
 CLA was used.

# Names should be added to this file like so:
#     Name <email address>
```

# Please keep the list sorted.

Damian Gryski <dgryski@gmail.com>
Jan Mercl <0xjnml@gmail.com>
Kai Backman <kaib@golang.org>
Marc-Antoine Ruel <maruel@chromium.org>
Nigel Tao <nigeltao@golang.org>
Rob Pike <r@golang.org>
Rodolfo Carvalho <rhcarvalho@gmail.com>
Russ Cox <rsc@golang.org>
Sebastien Binet <seb.binet@gmail.com>

1.251 groovy 2.4.12

1.252 reactive-streams 1.0.0

1.252.1 Available under license :

<WORK'S NAME> by <AUTHOR'S NAME>

To the extent possible under law, the person who associated CC0 with
<WORK'S NAME> has waived all copyright and related or neighboring rights
to <WORK'S NAME>.

You should have received a copy of the CC0 legalcode along with this
work. If not, see <http://creativecommons.org/publicdomain/zero/1.0/>.

1.253 swagger-annotations 1.5.16

1.253.1 Available under license :

```
/**
* Copyright 2016 SmartBear Software
* <p>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p>
* http://www.apache.org/licenses/LICENSE-2.0
* <p>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package io.swagger.annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
* License metadata available within the info section of a Swagger definition, see
* https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md#licenseObject
*
* @since 1.5.0
 */

@Target(ElementType.ANNOTATION_TYPE)
```

```
@Retention(RetentionPolicy.RUNTIME)
public @interface License {

  /**
   * The name of the license.
   *
   * @return the name of the license
   */
  String name();

  /**
   * An optional URL for the license.
   *
   * @return an optional URL for the license.
   */
  String url() default "";
}
```

1.254 olivere-elastic 6.1.27

1.254.1 Available under license :

```
# This is a list of people who have contributed code
# to the Elastic repository.
#
# It is just my small "thank you" to all those that helped
# making Elastic what it is.
#
# Please keep this list sorted.
```

0x6875790d0a [@huydx](https://github.com/huydx)
Adam Alix [@adamalix](https://github.com/adamalix)
Adam Weiner [@adamweiner](https://github.com/adamweiner)
Adrian Lungu [@AdrianLungu](https://github.com/AdrianLungu)
alehano [@alehano](https://github.com/alehano)
Alex [@akotlar](https://github.com/akotlar)
Alexander Sack [@asac](https://github.com/asac)
Alexandre Olivier [@aliphen](https://github.com/aliphen)
Alexey Sharov [@nizsheanez](https://github.com/nizsheanez)
AndreKR [@AndreKR](https://github.com/AndreKR)
Andr Bierlein [@ligustah](https://github.com/ligustah)
Andrew Dunham [@andrew-d](https://github.com/andrew-d)
Andrew Gaul [@andrewgaul](https://github.com/andrewgaul)
Andy Walker [@alaska](https://github.com/alaska)
Arquivei [@arquivei](https://github.com/arquivei)
arthurgustin
[@arthurgustin](https://github.com/arthurgustin)
Benjamin Fernandes [@LotharSee](https://github.com/LotharSee)
Benjamin Zarzycki [@kf6nux](https://github.com/kf6nux)
Boris Popovschi [@Zyqsempai](https://github.com/Zyqsempai)
Braden Bassingthwaite [@bbassingthwaite-va](https://github.com/bbassingthwaite-va)
Brady Love [@bradylove](https://github.com/bradylove)
Bryan Conklin [@bmconklin](https://github.com/bmconklin)
Bruce Zhou [@brucez-isell](https://github.com/brucez-isell)
Csar Jimnez [@cesarjimenez](https://github.com/cesarjimenez)
cforbes [@cforbes](https://github.com/cforbes)
Chris M [@tebriel](https://github.com/tebriel)
Chris Rice [@donutmonger](https://github.com/donutmonger)
Claudiu Olteanu [@claudiuolteanu](https://github.com/claudiuolteanu)
Chris Duncan [@veqryn](https://github.com/veqryn)
Christophe Courtaut [@kri5](https://github.com/kri5)
Connor Peet [@connor4312](https://github.com/connor4312)
Conrad Pankoff [@deoxxa](https://github.com/deoxxa)
Corey Scott [@corsc](https://github.com/corsc)
Daniel
Barrett [@shendaras](https://github.com/shendaras)
Daniel Heckrath [@DanielHeckrath](https://github.com/DanielHeckrath)
Daniel Imfeld [@dimfeld](https://github.com/dimfeld)
David Emanuel Buchmann [@wuurrd](https://github.com/wuurrd)
Dwayne Schultz [@myshkin5](https://github.com/myshkin5)
Ellison Leo [@ellisonleao](https://github.com/ellisonleao)
Erwin [@eticzon](https://github.com/eticzon)
Eugene Egorov [@EugeneEgorov](https://github.com/EugeneEgorov)
Evan Shaw [@edsrzf](https://github.com/edsrzf)
Fanfan [@wenpos](https://github.com/wenpos)
Faolan C-P [@fcheslack](https://github.com/fcheslack)
Filip Tepper [@filiptepper](https://github.com/filiptepper)

Gaspard Douady [@plopik](https://github.com/plopik)
Gaylord Aulke [@blafasel42](https://github.com/blafasel42)
Gerhard Hring [@ghaering](https://github.com/ghaering)
Guilherme Silveira [@guilherme-santos](https://github.com/guilherme-santos)
Guillaume J. Charmes [@creack](https://github.com/creack)
Guiseppe
[@gm42](https://github.com/gm42)
Han Yu [@MoonighT](https://github.com/MoonighT)
Harmen [@alicebob](https://github.com/alicebob)
Harrison Wright [@wright8191](https://github.com/wright8191)
Henry Clifford [@hcliff](https://github.com/hcliff)
Igor Dubinskiy [@idubinskiy](https://github.com/idubinskiy)
initialcontext [@initialcontext](https://github.com/initialcontext)
Isaac Saldana [@isaldana](https://github.com/isaldana)
J Barkey Wolf [@jjhbw](https://github.com/jjhbw)
Jack Lindamood [@cep21](https://github.com/cep21)
Jacob [@jdelgad](https://github.com/jdelgad)
Jayme Rotsaert [@jrots](https://github.com/jrots)
Jeff Rand [@jeffrand](https://github.com/jeffrand)
Jeremy Canady [@jrmycanady](https://github.com/jrmycanady)
Jrmie Vexiau [@texvex](https://github.com/texvex)
Jim Berlage [@jimberlage](https://github.com/jimberlage)
Joe Buck [@four2five](https://github.com/four2five)
John Barker [@j16r](https://github.com/j16r)
John Goodall [@jgoodall](https://github.com/jgoodall)
John Stanford
[@jxstanford](https://github.com/jxstanford)
Jonas Groenaas Drange [@semafor](https://github.com/semafor)
Josef Frhle [@Dexus](https://github.com/Dexus)
Josh Chorlton [@jchorl](https://github.com/jchorl)
jun [@coseyo](https://github.com/coseyo)
Junpei Tsuji [@jun06t](https://github.com/jun06t)
kartlee [@kartlee](https://github.com/kartlee)
Keith Hatton [@khatton-ft](https://github.com/khatton-ft)
kel [@liketic](https://github.com/liketic)
Kenta SUZUKI [@suzuken](https://github.com/suzuken)
Kevin Mulvey [@kmulvey](https://github.com/kmulvey)
Kyle Brandt [@kylebrandt](https://github.com/kylebrandt)
Leandro Piccilli [@lpic10](https://github.com/lpic10)
Lee [@leezhm](https://github.com/leezhm)
M. Zulfa Achsani [@misterciput](https://github.com/misterciput)
Maciej Lisiewski [@c2h5oh](https://github.com/c2h5oh)
Mara Kim [@autochthe](https://github.com/autochthe)
Marcy Buccellato [@marcybuccellato](https://github.com/marcybuccellato)
Mark Costello [@mcos](https://github.com/mcos)
Martin Hger
[@protomouse](https://github.com/protomouse)
Medhi Bechina [@mdzor](https://github.com/mdzor)

Mike Beshai [@mbesh](https://github.com/mbesh)
mmfrb [@mmfrb](https://github.com/mmfrb)
mnpritula [@mnpritula](https://github.com/mnpritula)
mosa [@mosasiru](https://github.com/mosasiru)
naimulhaider [@naimulhaider](https://github.com/naimulhaider)
Naoya Yoshizawa [@azihsoyn](https://github.com/azihsoyn)
navins [@ishare](https://github.com/ishare)
Naoya Tsutsumi [@tutuming](https://github.com/tutuming)
NeoCN [@NeoCN](https://github.com/NeoCN)
Nicholas Wolff [@nwolff](https://github.com/nwolff)
Nick K [@utrack](https://github.com/utrack)
Nick Whyte [@nickw444](https://github.com/nickw444)
Nicolae Vartolomei [@nvartolomei](https://github.com/nvartolomei)
Orne Brocaar [@brocaar](https://github.com/brocaar)
Paul [@eyeamera](https://github.com/eyeamera)
Pete C [@peteclark-ft](https://github.com/peteclark-ft)
Paolo [@ppiccolo](https://github.com/ppiccolo)
Radoslaw Wesolowski [r--w](https://github.com/r--w)
rchicoli
 [@rchicoli](https://github.com/rchicoli)
Roman Colohanin [@zuzmic](https://github.com/zuzmic)
Ryan Schmukler [@rschmukler](https://github.com/rschmukler)
Ryan Wynn [@rwynn](https://github.com/rwynn)
Sacheendra talluri [@sacheendra](https://github.com/sacheendra)
Sean DuBois [@Sean-Der](https://github.com/Sean-Der)
Shalin LK [@shalinlk](https://github.com/shalinlk)
singham [@zhaochenxiao90](https://github.com/zhaochenxiao90)
Slawomir CALUCH [@slawo](https://github.com/slawo)
Stephen Kubovic [@stephenkubovic](https://github.com/stephenkubovic)
Stuart Warren [@Woz](https://github.com/stuart-warren)
Sulaiman [@salajlan](https://github.com/salajlan)
Sundar [@sundarv85](https://github.com/sundarv85)
Swarlston [@Swarlston](https://github.com/Swarlston)
Take [ww24](https://github.com/ww24)
Tetsuya Morimoto [@t2y](https://github.com/t2y)
TimeEmit [@TimeEmit](https://github.com/timeemit)
TusharM [@tusharm](https://github.com/tusharm)
wangtuo [@wangtuo](https://github.com/wangtuo)
Wdney
 Yuri [@wedneyyuri](https://github.com/wedneyyuri)
wolfkdy [@wolfkdy](https://github.com/wolfkdy)
Wyndham Blanton [@wyndhblb](https://github.com/wyndhblb)
Yarden Bar [@ayashjorden](https://github.com/ayashjorden)
zakthomas [@zakthomas](https://github.com/zakthomas)
Yuya Kusakabe [@higebu](https://github.com/higebu)
Zach [@snowzach](https://github.com/snowzach)
zhangxin [@visaxin](https://github.com/visaxin)
@ [@zplzpl](https://github.com/zplzpl)

The MIT License (MIT)
Copyright 2012-2015 Oliver Eilhard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
 CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.255 cglib 3.2.0

1.255.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.256 okhttp 3.9.0

1.257 go-spew v1.1.1

1.257.1 Available under license :

ISC License

Copyright (c) 2012-2016 Dave Collins <dave@davec.name>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

1.258 go-resiliency v1.1.0

1.258.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Evan Huus

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN
 CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.259 go-xerial-snappy 776d571

1.259.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Evan Huus

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN
 CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.260 eapache-queue v1.1.0

1.260.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Evan Huus

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

1.261 pierrec-lz4 v2.0.7

1.261.1 Available under license :

1.262 cassandra-driver-core 3.0.2

1.263 classmate 0.8.0

1.264 hibernate-validator 5.0.0

1.265 jackson-dataformat-xml 2.7.9

1.265.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007. It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

1.266 jackson-dataformat-yaml 2.4.2

1.266.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
This copy of Jackson JSON processor YAML module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.267 jackson-datatype-joda 2.4.2

1.267.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.268 jackson-jaxrs-xml-provider 2.7.9

1.268.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.269 java-jwt 2.1.0

1.270 jboss-logging 3.1.1

1.270.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who
decide to use it. You
can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library,
whether gratis
or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Library,
and can be
reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary
GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding
machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the

source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that
uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the
object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the

copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse
you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system
which is
implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish
revised and/or new
versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision
will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE,
BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.271 jersey-bean-validation 2.5.1

1.272 jersey-client 2.5.1

1.273 jersey-common 2.5.1

1.274 jersey-container-servlet 2.5.1

1.275 jersey-container-servlet-core 2.5.1

1.276 jersey-media-multipart 2.5.1

1.277 jersey-server 2.5.1

1.278 metrics-json 3.1.0

1.279 minio 4.0.0

1.280 netty 4.1.27

1.281 spring-aspects 3.0.5

1.282 swagger-annotations 1.5.0

1.282.1 Available under license :

```
/**
* Copyright 2015 SmartBear Software
* <p/>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p/>
* http://www.apache.org/licenses/LICENSE-2.0
* <p/>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package io.swagger.annotations;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
* License metadata available within the info section of a Swagger definition, see
* https://github.com/swagger-api/swagger-spec/blob/master/versions/2.0.md#licenseObject
*
* @since 1.5.0
 */

@Target(ElementType.ANNOTATION_TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface License {

  /**
   * The name of the license.
   *
   * @return the name of the license
   */
  String name();
  /**
   * An optional URL for the license.
   *
   * @return an optional URL for the license.
   */
  String url() default "";
}
```

1.283 antlr 4.7.1

1.284 antlr-runtime 4.7.1

1.285 cglib 3.2.6

1.285.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.286 chill-java 0.7.4

1.287 classindex 3.4

1.288 commons-math 3.5

1.288.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License,
each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Math includes the following code provided to the ASF under the Apache License 2.0:

- The inverse error function implementation in the Erf class is based on CUDA code developed by Mike Giles, Oxford-Man Institute of Quantitative Finance, and published in GPU Computing Gems, volume 2, 2010 (grant received on March 23th 2013)
- The LinearConstraint, LinearObjectiveFunction, LinearOptimizer, RelationShip, SimplexSolver and SimplexTableau classes in package org.apache.commons.math3.optimization.linear include software developed by Benjamin McCann (http://www.benmccann.com) and distributed with the following copyright: Copyright 2009 Google Inc. (grant received on March 16th 2009)
- The class "org.apache.commons.math3.exception.util.LocalizedFormatsTest" which is an adapted version of "OrekitMessagesTest" test class for the Orekit library
- The "org.apache.commons.math3.analysis.interpolation.HermiteInterpolator"

has been imported from the Orekit space flight dynamics library.

===

APACHE COMMONS MATH DERIVATIVE WORKS:

The Apache commons-math library includes a number of subcomponents whose implementation is derived from original sources written in C or Fortran. License terms of the original sources are reproduced below.

===

For the lmder, lmpar and qrsolv Fortran routine from minpack and translated in the LevenbergMarquardtOptimizer class in package org.apache.commons.math3.optimization.general Original source copyright and license statement:

Copyright and license statement for the odex Fortran routine developed by E. Hairer and G. Wanner and translated in GraggBulirschStoerIntegrator class in package org.apache.commons.math3.ode.nonstiff:

===

Copyright and license statement for the original Mersenne twister C routines translated in MersenneTwister class in package org.apache.commons.math3.random:

===

The initial code for shuffling an array (originally in class "org.apache.commons.math3.random.RandomDataGenerator", now replaced by a method in class "org.apache.commons.math3.util.MathArrays") was inspired from the algorithm description provided in "Algorithms", by Ian Craw and John Pulham (University of Aberdeen 1999). The textbook (containing a proof that the shuffle is uniformly random) is available here:
http://citeseerx.ist.psu.edu/viewdoc/download;?doi=10.1.1.173.1898&rep=rep1&type=pdf

===

License statement for the direction numbers in the resource files for Sobol sequences.

Licence pertaining to sobol.cc and the accompanying sets of direction numbers

----

Copyright (c) 2008, Frances Y. Kuo and Stephen Joe
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the names of the copyright holders nor the names of the University of New South Wales and the University of Waikato and its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===

The initial commit of package "org.apache.commons.math3.ml.neuralnet" is
an adapted version of code developed in the context of the Data Processing
and Analysis Consortium (DPAC) of the "Gaia" project of the European Space
Agency (ESA).
===

The initial commit of the class "org.apache.commons.math3.special.BesselJ" is
an adapted version of code translated from the netlib Fortran program, rjbesl
http://www.netlib.org/specfun/rjbesl by R.J. Cody at Argonne National
Laboratory (USA). There is no license or copyright statement included with the
original Fortran sources.
===

The
 BracketFinder (package org.apache.commons.math3.optimization.univariate)
and PowellOptimizer (package org.apache.commons.math3.optimization.general)
classes are based on the Python code in module "optimize.py" (version 0.5)
developed by Travis E. Oliphant for the SciPy library (http://www.scipy.org/)
Copyright 2003-2009 SciPy Developers.

SciPy license
Copyright 2001, 2002 Enthought, Inc.
All rights reserved.

Copyright 2003-2013 SciPy Developers.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Enthought nor the names of the SciPy Developers may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY

1.289 disruptor 3.3.2

1.290 druid 1.0.15

1.291 elasticsearch-cli 6.3.1

1.291.1 Available under license :

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

1.292 elasticsearch-core 6.3.1

1.292.1 Available under license :

not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.293 elasticsearch-rest-client 6.3.1

1.293.1 Available under license :

Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

1.294 elasticsearch-x-content 6.3.1

1.294.1 Available under license :

Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.295 gbench 0.4.3

1.296 google-auth-library-credentials 0.4.0

1.297 grpc-auth 0.12.0

1.298 grpc-core 0.12.0

1.299 grpc-netty 0.12.0

1.300 grpc-stub 0.12.0

1.301 gson-fire 1.5.1

1.302 guice 4.2.0

1.302.1 Available under license :

Google Guice - Core Library
Copyright 2006-2018 Google, Inc.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

1.303 hpack 0.10.1

1.304 ical4j 1.0.2

1.305 icu4j 58.2

1.305.1 Available under license :

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

1. ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the
property of their respective owners.

2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

```
#     The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
#  The BSD License
#  http://opensource.org/licenses/bsd-license.php
 #  Copyright (C) 2006-2008, Google Inc.
#
#  All rights reserved.
#
#  Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
#  Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
#  Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
#  Neither the name of  Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
#
#  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
#
#
#  The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
#  * Libtabe (Chinese)
#    - https://sourceforge.net/project/?group_id=1519
#    - Its license terms and conditions are shown below.
#
#  * IPADIC (Japanese)
#    - http://chasen.aist-nara.ac.jp/chasen/distribution.html
 #    - Its license terms and conditions are shown below.
#
#  ---------COPYING.libtabe ---- BEGIN--------------------
#
#  /*
#   * Copyrighy (c) 1999 TaBE Project.
#   * Copyright (c) 1999 Pai-Hsiang Hsiao.
#   * All rights reserved.
#   *
#   * Redistribution and use in source and binary forms, with or without
#   * modification, are permitted provided that the following conditions
#   * are met:
#   *
#   * . Redistributions of source code must retain the above copyright
#   *   notice, this list of conditions and the following disclaimer.
#   * . Redistributions in binary form must reproduce the above copyright
#   *   notice, this list of conditions and the following disclaimer in
#   *   the documentation and/or other materials provided with the
#   *   distribution.
#   * . Neither the name of the TaBE Project nor the names of its
#   *   contributors may be used to endorse or promote products derived
#   *   from this software
 without specific prior written permission.
#   *
#   * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#   * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#   * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#   * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#   * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#   * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#   * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#   * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#   * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#   * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#   * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#   * OF THE POSSIBILITY OF SUCH DAMAGE.
#   */
#
#  /*
#   * Copyright (c) 1999 Computer Systems and Communication Lab,
#   *                    Institute
 of Information Science, Academia
#       *                    Sinica. All rights reserved.
#   *
#   * Redistribution and use in source and binary forms, with or without
#   * modification, are permitted provided that the following conditions
#   * are met:
#   *
#   * . Redistributions of source code must retain the above copyright
#   *   notice, this list of conditions and the following disclaimer.
#   * . Redistributions in binary form must reproduce the above copyright
#   *   notice, this list of conditions and the following disclaimer in
#   *   the documentation and/or other materials provided with the
#   *   distribution.
#   * . Neither the name of the Computer Systems and Communication Lab
#   *   nor the names of its contributors may be used to endorse or
#   *   promote products derived from this software without specific
#   *   prior written permission.
#   *
#   * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#   * "AS IS" AND
 ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#   * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#   * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#   * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#   * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#   * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#   * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#   * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#   * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#   * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#   * OF THE POSSIBILITY OF SUCH DAMAGE.
#   */
#
#  Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
#      University of Illinois
#  c-tsai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsai4
#
#  ---------------COPYING.libtabe-----END--------------------------------
#
#
#  ---------------COPYING.ipadic-----BEGIN-------------------------------
```

```
 #
#  Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
#  and Technology.  All Rights Reserved.
#
#  Use, reproduction, and distribution of this software is permitted.
#  Any copy of this software, whether in its original form or modified,
#  must include both the above copyright notice and the following
#  paragraphs.
#
#  Nara Institute of Science and Technology (NAIST),
#  the copyright holders, disclaims all warranties with regard to this
#  software, including all implied warranties of merchantability and
#  fitness, in no event shall NAIST be liable for
#  any special, indirect or consequential damages or any damages
#  whatsoever resulting from loss of use, data or profits, whether in an
#  action of contract, negligence or other tortuous action, arising out
#  of or in connection with the use or performance of this software.
#
#  A large portion of the dictionary entries
#  originate from ICOT Free
 Software.  The following conditions for ICOT
#  Free Software applies to the current dictionary as well.
#
#  Each User may also freely distribute the Program, whether in its
#  original form or modified, to any third party or parties, PROVIDED
#  that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
#  on, or be attached to, the Program, which is distributed substantially
#  in the same form as set out herein and that such intended
#  distribution, if actually made, will neither violate or otherwise
#  contravene any of the laws and regulations of the countries having
#  jurisdiction over the User or the intended distribution itself.
#
#  NO WARRANTY
#
#  The program was produced on an experimental basis in the course of the
#  research and development conducted during the project and is provided
#  to users as so produced on an experimental basis.  Accordingly, the
#  program is provided without any warranty whatsoever, whether express,
#  implied,
 statutory or otherwise.  The term "warranty" used herein
#  includes, but is not limited to, any warranty of the quality,
#  performance, merchantability and fitness for a particular purpose of
#  the program and the nonexistence of any infringement or violation of
#  any right of any third party.
#
#  Each user of the program will agree and understand, and be deemed to
#  have agreed and understood, that there is no warranty whatsoever for
#  the program and, accordingly, the entire risk arising from or
#  otherwise connected with the program is assumed by the user.
#
#  Therefore, neither ICOT, the copyright holder, or any other
#  organization that participated in or was otherwise related to the
#  development of the program and their respective officials, directors,
#  officers and other employees shall be held liable for any and all
#  damages, including, without limitation, general, special, incidental
#  and consequential damages, arising out of or otherwise in
 connection
#  with the use or inability to use the program or any product, material
#  or result produced or otherwise obtained by using the program,
#  regardless of whether they have been advised of, or otherwise had
#  knowledge of, the possibility of such damages at any time during the
#  project or thereafter.  Each user will be deemed to have agreed to the
#  foregoing by his or her commencement of use of the program.  The term
#  "use" as used herein includes, but is not limited to, the use,
#  modification, copying and distribution of the program and the
#  production of secondary products from the program.
#
#  In the case where the program, whether in its original form or
#  modified, was distributed or delivered to or received by a user from
#  any person, organization or entity other than ICOT, unless it makes or
#  grants independently of ICOT any specific warranty to the user in
#  writing, such person, organization or entity, will also be exempted
#  from
 and not be held liable to the user for any such damages as noted
#  above as far as the program is concerned.
#
#  ---------------COPYING.ipadic-----END----------------------------------
```

3. Lao Word Break Dictionary Data (laodict.txt)

```
#  Copyright (c) 2013 International Business Machines Corporation
#  and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# Dictionary: http://lao-dictionary.googlecode.com/git/Lao-Dictionary.txt
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
#              (copied below)
#
#  This file is derived from the above dictionary, with slight
#  modifications.
#  ----------------------------------------------------------------------
#  Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
#  All rights reserved.
#
#  Redistribution and use in source and binary forms, with or without
#  modification,
#  are permitted provided that the following conditions are met:
 #
#
# Redistributions of source code must retain the above copyright notice, this
#  list of conditions and the following disclaimer. Redistributions in
#  binary form must reproduce the above copyright notice, this list of
#  conditions and the following disclaimer in the documentation and/or
#  other materials provided with the distribution.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR
 TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
#  --------------------------------------------------------------------------
```

4. Burmese Word Break Dictionary Data (burmesedict.txt)

```
#  Copyright (c) 2014 International Business Machines Corporation
#  and others. All Rights Reserved.
#
#  This list is part of a project hosted at:
#    github.com/kanyawtech/myanmar-karen-word-lists
#
#  --------------------------------------------------------------------------
#  Copyright (c) 2013, LeRoy Benjamin Sharon
#  All rights reserved.
#
#  Redistribution and use in source and binary forms, with or without
#  modification, are permitted provided that the following conditions
#  are met: Redistributions of source code must retain the above
#  copyright notice, this list of conditions and the following
#  disclaimer.  Redistributions in binary form must reproduce the
#  above copyright
 notice, this list of conditions and the following
#  disclaimer in the documentation and/or other materials provided
#  with the distribution.
#
#    Neither the name Myanmar Karen Word Lists, nor the names of its
#    contributors may be used to endorse or promote products derived
#    from this software without specific prior written permission.
#
#  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
#  CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
#  INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
#  MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
#  DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
#  BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
#  EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
#  TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
#  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
#  ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 LIABILITY, OR
#  TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
#  THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
#  SUCH DAMAGE.
#  ------------------------------------------------------------------------
```

5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

```
# 7.  Database Ownership
#
#    The TZ database itself is not an IETF Contribution or an IETF
#    document.  Rather it is a pre-existing and regularly updated work
#    that is in the public domain, and is intended to remain in the
#    public domain.  Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do
#    not apply to the TZ Database or contributions that individuals make
#    to it.  Should any claims be made and substantiated against the TZ
#    Database, the organization that is providing
 the IANA
#    Considerations defined in this RFC, under the memorandum of
#    understanding with the IETF, currently ICANN, may act in accordance
#    with all competent court orders.  No ownership claims will be made
#    by ICANN or the IETF Trust on the database or the code.  Any person
#    making a contribution to the database or code waives all rights to
#    future claims in that contribution or in the TZ Database.
```

1.306 jcl-over-slf4j 1.7.19

1.307 jopt-simple 5.0.4

1.308 json-schema-validator 0.1.16

1.309 jsoup 1.9.2

1.310 jython 2.5.2

1.311 kafka-dropwizard-reporter 1.1.1

1.312 kryo 2.24.0

1.313 lang-mustache-client 6.3.1

1.313.1 Available under license :

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.314 mvel2 2.2.1

1.315 natty 0.13

1.316 netty 4.0.51

1.317 reindex-client 6.3.1

1.317.1 Available under license :

Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

1.318 remark 1.0.0

1.319 rhino 1.7.7.2

1.320 rocksdbjni 5.6.1

1.321 scala 2.11.11

1.322 siddhi-annotations 4.1.33

1.322.1 Available under license :

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.323 siddhi-core 4.1.33

1.323.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

1.324 siddhi-execution-regex 4.0.10

1.324.1 Available under license :

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

WSO2 Siddhi Execution Extension - Regex
Copyright 2017 WSO2

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.325 siddhi-execution-string 4.0.11

1.325.1 Available under license :

WSO2 Siddhi Execution String Extension
Copyright 2018 WSO2

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

1.326 siddhi-query-api 4.1.33

1.326.1 Available under license :

Siddhi Query API
Copyright 2018 WSO2

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.327 siddhi-query-compiler 4.1.33

1.327.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

1.328 slf4j 1.7.19

1.329 slf4j-ext 1.7.25

1.330 transport 6.3.1

1.330.1 Available under license :

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

1.331 transport-netty4-client 6.3.1

1.331.1 Available under license :

Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.332 zjsonpatch 0.3.6

1.332.1 Available under license :

```
package com.flipkart.zjsonpatch;

import com.fasterxml.jackson.databind.JsonNode;

class CopyingApplyProcessor extends InPlaceApplyProcessor {

   CopyingApplyProcessor(JsonNode target) {
       super(target.deepCopy());
   }
}
```

1.333 commons-http-client 3.1

1.334 d3 4.12.0

1.334.1 Available under license :

1.335 react-jsonschema-form 1.0.0

1.335.1 Available under license :

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.336 react-dates 15.2.0

1.336.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Airbnb

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.337 d3sankeydiagram 0.7.1

1.337.1 Available under license :

Copyright 2015, Mike Bostock
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

* Neither the name of the author nor the names of contributors may be used to
endorse or promote products derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS
BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.338 jackson 2.9.8

1.338.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.339 cxf 3.2.6

1.339.1 Available under license :

Apache CXF
Copyright 2006-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this

License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
/**
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements. See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership. The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
```

```
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/

package org.apache.cxf.io;

import java.io.IOException;
import java.io.InputStream;

/**
* Marker interface for OutputStreams that can directly support
* copying from an input stream.  OutputStreams
 that maintain their
* own byte buffer or similar may be able to optimize the copy
* instead of using the read/write into a temporary buffer that
* the normal IOUtils.copy method requires.
*/
public interface CopyingOutputStream {

   int copyFrom(InputStream in) throws IOException;

}
```

1.340 google-http-client 1.27.0

1.341 groovy-all 2.5.6

1.342 groovy-templates 2.5.6

1.343 groovy-console 2.5.6

1.344 groovy-xml 2.5.6

1.345 groovy-groovysh 2.5.6

1.346 bson 3.0.4

1.347 metrics-servlets 3.1.0

1.348 jackson-annotations 2.9.8

1.348.1 Available under license :

This copy of Jackson JSON processor annotations is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.349 sarama-shopify 1.19.0

1.349.1 Available under license :

Copyright (c) 2013 Shopify

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.350 aop-alliance 1.0

1.351 xpp3-min 1.1.4c

1.352 jquery 3.4.0

1.352.1 Available under license :

Copyright JS Foundation and other contributors, https://js.foundation/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The BSD License)

Copyright (c) 2010-2017, Christian Johansen, christian@cjohansen.no
All rights reserved.

BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules directory are externally maintained
libraries used by this software which have their own licenses; we
recommend you read them, as their terms may differ from the terms above.
Copyright JS Foundation and other contributors, https://js.foundation/

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/jquery/sizzle

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from

the terms above.

1.353 cxf-rt-frontend-jaxrs 3.2.6

1.353.1 Available under license :

Apache CXF
Copyright 2006-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

1.354 paesslerag-jsonpath 0.1.1

1.354.1 Available under license :

1.355 jackson 2.7.9

1.356 jackson-core-asl 1.9.13

1.357 woodstox-core-asl 4.4.1

1.358 snappy-java 1.1.7.1

1.359 jcip-annotation 1.0

1.360 commons-cli 1.4

1.360.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.361 zkclient 0.10

1.362 c3p0 0.9.1.1

1.362.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software

packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these

rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that

there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and

is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the
reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU

operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by
the copyright holder or
other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code"
for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided
that
you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to

exercise the right to control the distribution
of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for

that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties
are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the
Library even though the source code is not.
Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable
source code for the Library including whatever
changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified

executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise

permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However,
parties who have received copies,
or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library

subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly

through you, then
the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing
to
distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version,
but
may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is

copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free
status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

1.363 jcommander 1.48

1.364 smackjava 4.1.9

1.365 javassist 3.21.0

1.366 httpcomponents-client 4.5.2

1.366.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

===

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <http://mozilla.org/MPL/2.0/>

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code

Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of
version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently,
any and
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to,
deletion from, or modification of the contents of Covered
Software; or

(b) any new file in Source Code Form that contains any Covered
Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method,
process, and apparatus claims, in any patent Licensable by such
Contributor that would be infringed, but for the grant of the
License, by the making, using, selling, offering for sale, having
made, import, or transfer of either its Contributions or its
Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU
Lesser General Public License, Version 2.1, the GNU Affero General
Public License, Version 3.0, or any later versions of those

licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications
of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have
under
applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered

Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability
 terms You offer. You may include additional
disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However,
 if You become
compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the
 rights granted to
You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all
end user license agreements (excluding distributors and resellers) which
have been validly granted by You or Your distributors under this License
prior to termination shall survive termination.

```
************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is
 free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*
                                                                     *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation, damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation
 of          *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                    *
```

*   *
****

8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation
 which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a
distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove
any references to the name of the license

steward (except to note that
such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary
Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary
Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

1.367 commons-io 2.4

1.367.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant
of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

1.368 json-path 2.2.0

1.368.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/InvalidPathException.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/JsonPath.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/Option.java
*
/opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/CompiledPath.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/PredicateContextImpl.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/Utils.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-

jar/com/jayway/jsonpath/internal/JsonFormatter.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/json/JsonProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/mapper/JsonSmartMappingProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/JsonContext.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/ValueCompareException.java
*
 /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/json/JacksonJsonProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/mapper/GsonMappingProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/Predicate.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/DocumentContext.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/PropertyPathToken.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/mapper/TapestryMappingProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/mapper/MappingException.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/EvaluationContextImpl.java
*
 /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/mapper/JacksonMappingProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/PathNotFoundException.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/json/AbstractJsonProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/InvalidModificationException.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/PathToken.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/PredicatePathToken.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/json/GsonJsonProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/EvaluationListener.java
*
 /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/MapFunction.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/mapper/MappingProvider.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-

jar/com/jayway/jsonpath/WriteContext.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/Criteria.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/EvaluationContext.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/ReadContext.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/TypeRef.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/RootPathToken.java
*
/opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/Configuration.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/Filter.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/ScanPathToken.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/cache/LRUCache.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/ParseContext.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/WildcardPathToken.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/InvalidCriteriaException.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/path/ArrayPathToken.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/spi/json/JsonSmartJsonProvider.java
*
/opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/InvalidJsonException.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/internal/Path.java
* /opt/cola/permits/1000738790_1651084395.64/0/json-path-2-2-0-sources-jar/com/jayway/jsonpath/JsonPathException.java

1.369 jackson-annotations 2.8.0

1.369.1 Available under license :

This copy of Jackson JSON processor annotations is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.370 commons-io 1.3.2

1.370.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)

the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication
on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant
of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate

as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache Jakarta Commons IO
Copyright 2001-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.371 scala-reflect 2.11.12

1.372 httpcomponents-client 4.2.5

1.372.1 Available under license :

Apache HttpComponents Client
Copyright 1999-2012 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.373 httpcomponents-core 4.2.4

1.373.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

===

This project contains annotations in the package org.apache.http.annotation
which are derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Full text: http://creativecommons.org/licenses/by/2.5/legalcode

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS
PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR

OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE
LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose
of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the
copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

to create and reproduce Derivative Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket

Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created
by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise
of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display,
publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author,"
or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on
Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Apache HttpComponents HttpCore
Copyright 2005-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

1.374 smack-tcp 4.1.9

1.375 smack-resolver-javax 4.1.9

1.376 smack-im 4.1.9

1.377 smack-sasl-javax 4.1.9

1.378 smack-extensions 4.1.9

1.379 sprintf-js 1.0.3

1.379.1 Available under license :

Copyright (c) 2007-2014, Alexandru Marasteanu <hello [at) alexei (dot] ro>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright
 notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions and the following disclaimer in the
 documentation and/or other materials provided with the distribution.
* Neither the name of this software nor the names of its contributors may be

used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.380 javax-annotation-api 1.2

1.380.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8.
Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing

Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12.
Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under
intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software
with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the
combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available
in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices
contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee

for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT

WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES
THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses
that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS
DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item,. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software,
 we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer
 you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any
 program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute
verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that
in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works
in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or
executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code
for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute
the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives
a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only

way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear
what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These
Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits
your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular
source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but
you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

1.381 grpc-grpc-common 1.0

1.381.1 Available under license :

Copyright (c) 2015, grpc
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this

1.382 jackson-databind 2.9.9

1.382.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.383 lodash 4.17.11

1.383.1 Available under license :

Software License Agreement (BSD License)

Copyright (c) 2007, Parakey Inc.
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above
 copyright notice, this list of conditions and the
 following disclaimer.

* Redistributions in binary form must reproduce the above
 copyright notice, this list of conditions and the
 following disclaimer in the documentation and/or other
 materials provided with the distribution.

* Neither the name of Parakey Inc. nor the names of its
 contributors may be used to endorse or promote products
 derived from this software without specific prior
 written permission of Parakey Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS
 FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT

OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright (c) 2010-2016 Jeremy Ashkenas, DocumentCloud

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2009-2016 Jeremy Ashkenas, DocumentCloud and Investigative
Reporters & Editors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT
OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.384 rabbitmq-java-client 3.6.2

1.384.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed
to make sure that you
have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which
contains
a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies
of the Program's
source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole
or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered
independent and separate works in
themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or
executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the
source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such

parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through

any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number
of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN
IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
   <one line to give the program's name and a brief idea of what it does.>
   Copyright (C) <year>  <name of author>

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 2 of the License,
 or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License along
   with this program; if not, write to the Free Software Foundation, Inc.,
   51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
  This is free software, and you are welcome to redistribute it
  under certain conditions; type `show c' for details.
```

The hypothetical commands `show w'
 and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

 Yoyodyne, Inc., hereby disclaims all copyright interest in the program
 `Gnomovision' (which makes passes at compilers) written by James Hacker.

 <signature of Ty Coon>, 1 April 1989
 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of

version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently, any and
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer
for sale, have made, import, and otherwise transfer either its
Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the

Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional
 Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the
 maximum extent possible; and (b)
describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
************************************************************************
*
                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered
 Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                *
*  --------------------------                                *
*                                                            *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation,
 damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation of           *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                *
*                                                            *
************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place
 of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version
number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source
Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this
file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

This package, the RabbitMQ Java client library, is triple-licensed under the Mozilla Public License 2.0 ("MPL"), the GNU General Public License version 2 ("GPL") and the Apache License version 2 ("ASL"). For the MPL, please see LICENSE-MPL-RabbitMQ. For the GPL, please see LICENSE-GPL2. For the ASL, please see LICENSE-APACHE2.

This means that you may choose one of these licenses when including or using this software in your own.

The RabbitMQ Java client library includes third-party software under the ASL. For this license, please see LICENSE-APACHE2. For attribution of copyright and other details of provenance, please refer to the source code.

If you have any questions regarding licensing, please contact us at info@rabbitmq.com.

1.385 jsr305 1.3.9

1.385.1 Available under license :

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

1.386 commons-lang3 3.4

1.386.1 Available under license :

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

1.387 socket.io-client 2.0.1

1.387.1 Available under license :

1.388 goavro 2.7.0

1.388.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate

1.389 tsenart-tb 19f4c3d

1.389.1 Available under license :

No license file was found, but licenses were detected in source scan.

# Token Bucket (tb) [![Build Status](https://secure.travis-ci.org/tsenart/tb.png)](http://travis-ci.org/tsenart/tb)
[![GoDoc](https://godoc.org/github.com/tsenart/tb?status.png)](https://godoc.org/github.com/tsenart/tb)

This package provides a generic lock-free implementation of the "Token bucket"
algorithm where handling of non-conformity is left to the user.

> The token bucket is an algorithm used in packet switched computer networks and telecommunications networks. It can be used to check that data transmissions, in the form of packets, conform to defined limits on bandwidth and burstiness (a measure of the unevenness or variations in the traffic flow)
-- <cite>[Wikipedia](http://en.wikipedia.org/wiki/Token_bucket)</cite>

This implementation of the token bucket generalises its applications beyond packet rate conformance. Hence, the word *generic*. You can use it to throttle any flow over time as long as it can
be expressed as a number (bytes/s, requests/s, messages/s, packets/s, potatoes/s, heartbeats/s, etc...).

The *lock-free* part of the description refers to the lock-free programming techniques (CAS loop) used in the core `Bucket` methods (`Take` and `Put`). [Here is](http://preshing.com/20120612/an-introduction-to-lock-free-programming/) a good overview of lock-free programming you can refer to.

All utility pacakges such as http and [io](io/) are just wrappers around the core package.
This ought to be your one stop shop for all things **throttling** in Go so feel free to propose missing common functionality.

## Install
```shell
$ go get github.com/tsenart/tb
```

## Usage
Read up the [docs](https://godoc.org/github.com/tsenart/tb) and have a look at some examples.

## Licence
```
The MIT License (MIT)

Copyright (c) 2014-2015 Toms Senart

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation
 files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Found in path(s):
* /opt/cola/permits/1112656902_1607366476.07/0/tsenart-tb-zip/tb-master/README.md

1.390 socket.io 2.0.1

1.390.1 Available under license :

(The MIT License)

Copyright (c) 2014-2017 Automattic <dev@cloudup.com>

Permission is hereby granted, free of charge, to any person obtaining

1.391 commons-lang3 3.6

1.391.1 Available under license :

Apache Commons Lang
Copyright 2001-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.

1.392 bean-validation-api 1.1.0

1.392.1 Available under license :

1.393 flink-metrics-core 1.4.0

1.393.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

flink-metrics-core
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.394 jaeger-core 0.23.0

1.394.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2016-2017, Uber Technologies, Inc
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/RemoteReporter.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2017, Uber Technologies, Inc
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
```

* the License.
*/

Found in path(s):
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/Reference.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/HexCodec.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/BaggageRestrictionManagerProxy.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/HttpBaggageRestrictionManagerProxy.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/BaggageSetter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/utils/Java6CompatibleThreadLocalRandom.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/BaggageRestrictionManager.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/BaggageRestrictionManagerException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/RemoteBaggageRestrictionManager.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/B3TextMapCodec.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/Restriction.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/DefaultBaggageRestrictionManager.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/baggage/http/BaggageRestrictionResponse.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2016, Uber Technologies, Inc
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/Counter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/Injector.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/StatsFactory.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/utils/SystemClock.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/PerOperationSampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/SpanContext.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/Extractor.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/SamplingStatus.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/http/ProbabilisticSamplingStrategy.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/LoggingReporter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/NoopReporter.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/utils/Utils.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/ConstSampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/EmptyIpException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/NotFourOctetsException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/UnknownSamplingStrategyException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/RateLimitingSampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/senders/UdpSender.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/InMemoryStatsReporter.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/Reporter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/SamplingManager.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/Configuration.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-

jar/com/uber/jaeger/samplers/ProbabilisticSampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/GuaranteedThroughputSampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/RemoteControlledSampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/NullStatsReporter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/Timer.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/SenderException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/utils/Http.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/StatsFactoryImpl.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/Gauge.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/http/SamplingStrategyResponse.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/Span.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/MalformedTracerStateStringException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/senders/ThriftSender.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/Metric.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/protocols/JaegerThriftSpanConverter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/http/OperationSamplingParameters.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/Tag.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/LogData.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/InMemoryReporter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/utils/RateLimiter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/UnsupportedFormatException.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/http/RateLimitingSamplingStrategy.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/StatsReporter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/EmptyTracerStateStringException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/utils/Clock.java

* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/Constants.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/PrefixedKeys.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/TextMapCodec.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/senders/HttpSender.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/CompositeReporter.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/http/PerOperationSamplingParameters.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/Tracer.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/Sampler.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/metrics/Metrics.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/senders/Sender.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/exceptions/SamplingStrategyErrorException.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/reporters/protocols/ThriftUdpTransport.java
*
 /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/samplers/HttpSamplingManager.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/Codec.java
* /opt/cola/permits/1112657307_1607127890.0/0/jaeger-core-0-23-0-sources-3-jar/com/uber/jaeger/propagation/CompositeCodec.java

1.395 jsr305 3.0.1

1.395.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2005 Brian Goetz
* Released under the Creative Commons Attribution License
*   (http://creativecommons.org/licenses/by/2.5)
* Official home: http://www.jcip.net
*/
```

Found in path(s):
* /opt/cola/permits/1112657325_1607127883.66/0/jsr305-3-0-1-sources-3-jar/javax/annotation/concurrent/GuardedBy.java
* /opt/cola/permits/1112657325_1607127883.66/0/jsr305-3-0-1-sources-3-jar/javax/annotation/concurrent/Immutable.java
* /opt/cola/permits/1112657325_1607127883.66/0/jsr305-3-0-1-sources-3-jar/javax/annotation/concurrent/NotThreadSafe.java

1.396 google-http-client-xml 1.20.0

1.396.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (c) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/http/xml/atom/package-info.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/http/xml/atom/AtomContent.java

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/package-info.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/atom/package-info.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/atom/AbstractAtomFeedParser.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/GenericXml.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/XmlNamespaceDictionary.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/http/xml/package-info.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/atom/Atom.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/http/xml/AbstractXmlHttpContent.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/http/xml/atom/AtomFeedParser.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/http/xml/XmlHttpContent.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/XmlObjectParser.java
* /opt/cola/permits/1112657277_1607127823.14/0/google-http-client-xml-1-20-0-sources-1-jar/com/google/api/client/xml/Xml.java

1.397 flink-connector-kafka 2.11-1.4.0

1.397.1 Available under license :

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

flink-connector-kafka
Copyright 2014-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.398 scala-logging_2.12 3.7.2

1.398.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.399 commons-lang3 3.5

1.399.1 Available under license :

Apache Commons Lang
Copyright 2001-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.400 apache-log4j 2.8

1.400.1 Available under license :

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this

License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.401 jaeger-thrift 0.23.0

1.401.1 Available under license :

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the

content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.402 commons-lang3 2.6

1.402.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant

of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache Commons Lang
Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

1.403 commons-lang3 3.2.1

1.403.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Lang
Copyright 2001-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

1.404 commons-lang3 3.3.1

1.404.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Apache Commons Lang
Copyright 2001-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

1.405 groovy-sql 2.4.6

1.405.1 Available under license :

No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/package.html
No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one
*  or more contributor license agreements.  See the NOTICE file
*  distributed with this work for additional information
*  regarding copyright ownership.  The ASF licenses this file
*  to you under the Apache License, Version 2.0 (the
```

Found in path(s):
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/BatchingStatementWrapper.java
*
 /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/CallResultSet.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/ResultSetOutParameter.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/Sql.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/GroovyRowResult.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/SqlWhereVisitor.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/InOutParameter.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/SqlOrderByVisitor.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/BatchingPreparedStatementWrapper.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/DataSet.java
*
 /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/ExpandedVariable.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/ExtractIndexAndSql.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/ResultSetMetaDataWrapper.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/SqlWithParams.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/InParameter.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/GroovyResultSet.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/GroovyResultSetProxy.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/GroovyResultSetExtension.java
* /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/org/codehaus/groovy/runtime/SqlGroovyMethods.java
*
 /opt/cola/permits/1112657301_1607127833.42/0/groovy-sql-2-4-6-sources-1-jar/groovy/sql/OutParameter.java

1.406 slf4j-log4j 1.7.19

1.406.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/**
* Copyright (c) 2004-2011 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free  of charge, to any person obtaining
* a  copy  of this  software  and  associated  documentation files  (the
* "Software"), to  deal in  the Software without  restriction, including
* without limitation  the rights to  use, copy, modify,  merge, publish,
* distribute,  sublicense, and/or sell  copies of  the Software,  and to
* permit persons to whom the Software  is furnished to do so, subject to
* the following conditions:
*
* The  above  copyright  notice  and  this permission  notice  shall  be
* included in all copies or substantial portions of the Software.
*
* THE  SOFTWARE IS  PROVIDED  "AS  IS", WITHOUT  WARRANTY  OF ANY  KIND,
* EXPRESS OR  IMPLIED, INCLUDING  BUT NOT LIMITED  TO THE  WARRANTIES OF
* MERCHANTABILITY,    FITNESS    FOR    A    PARTICULAR    PURPOSE    AND
* NONINFRINGEMENT. IN
 NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE,  ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*
*/
```

Found in path(s):

```
* /opt/cola/permits/1112657313_1607127856.17/0/slf4j-log4j12-1-7-19-sources-2-jar/org/slf4j/impl/Log4jLoggerAdapter.java
* /opt/cola/permits/1112657313_1607127856.17/0/slf4j-log4j12-1-7-19-sources-2-jar/org/slf4j/impl/StaticMDCBinder.java
* /opt/cola/permits/1112657313_1607127856.17/0/slf4j-log4j12-1-7-19-sources-2-jar/org/slf4j/impl/Log4jLoggerFactory.java
* /opt/cola/permits/1112657313_1607127856.17/0/slf4j-log4j12-1-7-19-sources-2-jar/org/slf4j/impl/StaticLoggerBinder.java
* /opt/cola/permits/1112657313_1607127856.17/0/slf4j-log4j12-1-7-19-sources-2-jar/org/slf4j/impl/Log4jMDCAdapter.java
* /opt/cola/permits/1112657313_1607127856.17/0/slf4j-log4j12-1-7-19-sources-2-jar/org/slf4j/impl/StaticMarkerBinder.java
```

1.407 flink-core 1.4.0

1.407.1 Available under license :

revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

flink-core
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.408 commons-codec 1.9

1.408.1 Available under license :

Apache Commons Codec
Copyright 2002-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from)
the Work and for which the
editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

1.409 error_prone_annotations 2.1.3

1.409.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean
the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name)
to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual
or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by
combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.410 j2objc-annotations 1.1

1.410.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean
the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name)
to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual
or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by
combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that
such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any
Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims
asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.411 hppc 0.7.1

1.411.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2010-2013, Carrot Search s.c., Boznicza 11/56, Poznan, Poland

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the

1.412 woodstox-core 5.1.0

1.412.1 Available under license :

1.413 commons-logging 1.1.1

1.413.1 Available under license :

otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
// --
// NOTICE file corresponding to the section 4d of The Apache License,

// Version 2.0, in this case for Commons Logging
// --

Commons Logging
Copyright 2001-2007 The Apache Software Foundation

This product includes/uses software(s) developed by 'an unknown organization'
- Unnamed - avalon-framework:avalon-framework:jar:4.1.3
- Unnamed - log4j:log4j:jar:1.2.12
- Unnamed - logkit:logkit:jar:1.0.1

1.414 commons-cli 1.3.1

1.414.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only

Apache Commons CLI
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.415 jackson-jaxrs 2.7.9

1.415.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.416 servlet-api 3.1.0

1.416.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8.

Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12.

Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under

intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software
with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the
combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available
in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES
THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer
or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses
that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION

OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS
DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item,. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this
License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for
compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California
and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software,
we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer
you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any

program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute
verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that
in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works
in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE

STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These
Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits
your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular
source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but
you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

1.417 google-http-client-jackson2 1.22.0

1.417.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1517726816_1671591092.1641777/0/google-http-client-jackson2-1-22-0-sources-4-jar/com/google/api/client/json/jackson2/JacksonGenerator.java
* /opt/cola/permits/1517726816_1671591092.1641777/0/google-http-client-jackson2-1-22-0-sources-4-jar/com/google/api/client/json/jackson2/JacksonParser.java
* /opt/cola/permits/1517726816_1671591092.1641777/0/google-http-client-jackson2-1-22-0-sources-4-jar/com/google/api/client/json/jackson2/package-info.java
*
/opt/cola/permits/1517726816_1671591092.1641777/0/google-http-client-jackson2-1-22-0-sources-4-jar/com/google/api/client/json/jackson2/JacksonFactory.java

1.418 google-http-client-jackson2 1.20.0

1.418.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.

*/

Found in path(s):
* /opt/cola/permits/1517726831_1671584444.6829457/0/google-http-client-jackson-1-20-0-sources-3-jar/com/google/api/client/json/jackson/package-info.java
* /opt/cola/permits/1517726831_1671584444.6829457/0/google-http-client-jackson-1-20-0-sources-3-jar/com/google/api/client/json/jackson/JacksonGenerator.java
* /opt/cola/permits/1517726831_1671584444.6829457/0/google-http-client-jackson-1-20-0-sources-3-jar/com/google/api/client/json/jackson/JacksonFactory.java
*
 /opt/cola/permits/1517726831_1671584444.6829457/0/google-http-client-jackson-1-20-0-sources-3-jar/com/google/api/client/json/jackson/JacksonParser.java

1.419 jackson-xc 2.7.9

1.419.1 Available under license :

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.420 elasticsearch-secure-sm 6.3.1

1.420.1 Available under license :

Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.421 jackson-mapper-asl 1.9.13-ASL License

1.421.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first
think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program.
We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom.
The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In

other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This

License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means

the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice

and disclaimer of warranty; keep
intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the
facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same
sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General
Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including

whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable
linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.
For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that
do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
8. You may not copy, modify, sublicense, link with,
or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the

recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies
directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This
section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later
version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF
THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.
Copyright (C) year name of author

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation;
either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public

License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in
the library 'Frob' (a library for tweaking knobs) written
by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon,
President of Vice
That's all there is to it!

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii)
beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own

copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names
of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law
(such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
/* Jackson JSON-processor.
*
* Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi
*
* Licensed under the License specified in file LICENSE, included with
* the source code and binary code bundles.
* You may not use this file except in compliance with the License.
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
```

*/