Cisco IOS Debug Command Reference - Commands M through R

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

debug management event through debug mpls ldp bindings

debug management event mib

To monitor the activities of the Event MIB in real time on your routing device, use the **debug management event mib**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug management event mib
no debug management event mib

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.

Usage Guidelines

The **debug management event mib** command prints messages to the screen whenever the Event MIB evaluates a specified trigger. These messages are given in real-time, and are intended to be used by technical support engineers for troubleshooting purposes. Definitions for the OID (object identifier) fields can be found in the EVENT-MIB.my file, available for download from the Cisco MIB website on http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

Examples

The following is sample output from the **debug management event mib**command:

```
Router# debug management event mib
Event Process Bool: Owner aseem, Trigger 01
 Event Bool process: invoke event
 Event Bool process: no wildcarding
Event: OID ifEntry.10.3
Event getValue abs: 69847284
 Event Bool process: Trigger Fired !
 mteSetNotifyObjects:
 Event execOnFiring: sending notification
Event: OID ifEntry.10.1
Event add_objects: Owner , Trigger
Event add_objects: Owner aseem, Trigger sethi
Event Found Owner: aseem
Event Found Name: sethi
Event: OID ifEntry.10.1
 Event: sending trap with 7 OIDs
Event: OID mteHotTrigger.0
Event: OID mteHotTargetName.0
Event: OID mteHotContextName.0
Event: OID ifEntry.10.3
Event: OID mteHotValue.0
Event: OID ifEntry.10.1
Event: OID ifEntry.10.1
```

```
Event mteDoSets: setting oid
 Event mteDoSets: non-wildcarded oid
Event: OID ciscoSyslogMIB.1.2.1.0
Event Thresh Process: Owner aseem, Trigger 01
 Event Thresh process: invoke rising event
 Event Thresh process: invoke falling event
 Event Thresh process: no wildcarding
Event: OID ifEntry.10.3
Event getValue abs: 69847284
Event Existence Process: Owner aseem, Trigger 01
 Event Exist process: invoke event
 Event Exist process: no wildcarding
Event: OID ifEntry.10.3
Event getValue abs: 69847284
 Event Check ExistTrigger for Absent
 Event Check ExistTrigger for Changed
Router# no debug management event mib
```

Related Commands

Command	Description
show management event	Displays the SNMP Event values that have been configured on your routing device through the use of the Event MIB.

debug management expression

To monitor the activities of the Expression MIB in real time on your routing device, use the **debug management expression**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug management expression {**evaluator** | **mib** | **parser**}
debug management expression {**evaluator** | **mib** | **parser**}

Syntax Description

evaluator	Specifies the Expression MIB evaluator.
mib	Specifies the Expression MIB SNMP operations.
parser	Specifies the Expression MIB parsing.

Command Default

By default, debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(1)	This command was introduced in a release earlier than Cisco IOS Release 12.2(1).
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2SR	This command is supported in the Cisco IOS Release 12.2SR train. Support in a specific 12.2SR release of this train depends on your feature set, platform, and platform hardware.
12.2SB	This command is supported in the Cisco IOS Release 12.2SB train. Support in a specific 12.2SB Release of this train depends on your feature set, platform, and platform hardware.

Examples

The following example shows how to enable debugging options for Expression MIB:

```
Router# debug management expression mib
Expression MIB SNMP operations debugging is on
```

Related Commands

Command	Description
show management expression	Displays the SNMP Expression values that have been configured on your routing device through the use of the Expression MIB.

debug mdns

To enable the debugging of multicast Domain Name System (mDNS) service discovery information, use the **debug mdns** command in privileged EXEC mode. To disable the debugging output, use the **no** form of this command.

debug mdns {**all** | **error** | **event** | **packet** | **verbose**}
no debug mdns

Syntax Description

all	Enables logging of the information about the mDNS service discovery processes.
error	Enables logging of the information about the errors encountered by the mDNS responder.
event	Enables logging of the information about the various events such as free, memory-allocated, packet, request, and timer.
packet	Enables logging of the information about the hex dump (byte by byte printing of packet traffic information) moving in and out of the mDNS responder.
verbose	Enables logging of detailed mDNS service discovery information.

Command Default

Debugging of mDNS service discovery is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.3(2)S	This command was introduced.

Examples

The following example shows how to enable debugging output for mDNS events:

```
Device> enable
Device# debug mdns event
Device# mDNS event debugs debugging is on
Device# sh log
Syslog logging: enabled (0 messages dropped, 14 messages rate-limited, 0 flushes, 0 overruns,
 xml disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 4762561 messages logged, xml disabled,
                    filtering disabled
```

```
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 176 message lines logged
        Logging Source-Interface:       VRF Name:
Log Buffer (4096 bytes):
er cache hit!
*Mar 15 03:01:38.234: SISF[CLA]: Interested feature:
*Mar 15 03:01:38.234: SISF[CLA]:                Snooping
*Mar 15 03:01:38.234: SISF[SWI]: Gi0/0/1 vlan 0 Feature_0  Snooping priority 128
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0 Parse msg ND_ROUTER_ADVERT. len 48
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0 Found 3 options
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0           option 1 : ND_OPT_SOURCE_LINKADDR
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0             option 3 :
ND_OPT_PREFIX_INFORMATION
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0             option 5 : ND_OPT_MTU
*Mar 15 03:01:38.234: SISF[PRS]:
*Mar 15 03:01:38.234: SISF[GLN]: Gi0/0/1 vlan 0 IPv6 snooping Gleaner setting sec level to
 2
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0 Sec level is Guard
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0 Advertise from access: default action is
update entry
*Mar 15 03:01:38.234: SISF[PRS]: Gi0/0/1 vlan 0 Unallowed RA/Redir: default action is delete
 entry
*Mar 15 03:01:38.234: SISF[GLN]: Gi0/0/1 vlan 0 Unauthorized packet
*Mar 15 03:01:38.234: SISF[SWI]: Gi0/0/1 vlan 0 Feature Snooping rc 1
*Mar 15 03:01:38.235: SISF[SWI]: Gi0/0/1 vlan 0 Feature drop
*Mar 15 03:01:38.235: SISF[MEM]: Unlocking, count is now 0
*Mar 15 03:01:38.235: SISF[MEM]:  3BB56338 semaphore system unlocked
*Mar 15 03:01:38.485: SISF[SWI]: SISF IPv6 enqueue FE80::217:95FF:FE73:9600
*Mar 15 03:01:40.716: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:01:40.866: SISF[SWI]: SISF IPv6 enqueue FE80::213:80FF:FE3E:8B25
*Mar 15 03:01:41.466: SISF[SWI]: SISF IPv6 enqueue FE80::213:80FF:FE3E:8B24
*Mar 15 03:01:41.644: SISF[SWI]: SISF IPv6 enqueue FE80::221:D8FF:FECD:5F40
*Mar 15 03:01:45.376: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:01:49.732: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:01:50.463: SISF[SWI]: Match ACL for incoming packet on Gi0/0/1
*Mar 15 03:01:50.463: SISF[SWI]: SISF IPv6 highjack L3-IF Gi0/0/1
*Mar 15 03:01:50.463: SISF[MEM]: Owner is this process
*Mar 15 03:01:50.463: SISF[MEM]: semaphore 3BB56338 (re)locked
*Mar 15 03:01:50.463: SISF[MEM]: Locking, count is now 1
*Mar 15 03:01:50.463: SISF[CLA]: Building interested feature list
*Mar 15 03:01:50.463: SISF[CLA]: Interest on target Gi0/0/1
*Mar 15 03:01:50.463: SISF[CLA]: Classifier cache hit!
*Mar 15 03:01:50.463: SISF[CLA]: Interested feature:
*Mar 15 03:01:50.463: SISF[CLA]:                Snooping
*Mar 15 03:01:50.463: SISF[SWI]: Gi0/0/1 vlan 0 Feature_0  Snooping priority 128
*Mar 15 03:01:50.463: SISF[PRS]: Gi0/0/1 vlan 0 Parse msg ND_ROUTER_ADVERT. len 48
*Mar 15 03:01:50.463: SISF[PRS]: Gi0/0/1 vlan 0 Found 3 options
*Mar 15 03:01:50.463: SISF[PRS]: Gi0/0/1 vlan 0           option 1 : ND_OPT_SOURCE_LINKADDR
*Mar 15 03:01:50.463: SISF[PRS]: Gi0/0/1 vlan 0             option 3 :
ND_OPT_PREFIX_INFORMATION
*Mar 15 03:01:50.463: SISF[PRS]: Gi0/0/1 vlan 0             option 5 : ND_OPT_MTU
*Mar 15 03:01:50.464: SISF[PRS]:
*Mar 15 03:01:50.464: SISF[GLN]: Gi0/0/1 vlan 0 IPv6 snooping Gleaner setting sec level to
 2
*Mar 15 03:01:50.464: SISF[PRS]: Gi0/0/1 vlan 0 Sec level is Guard
*Mar 15 03:01:50.464: SISF[PRS]: Gi0/0/1 vlan 0 Advertise from access: default action is
update entry
*Mar 15 03:01:50.464: SISF[PRS]: Gi0/0/1 vlan 0 Unallowed RA/Redir: default action is delete
 entry
```

```
*Mar 15 03:01:50.464: SISF[GLN]: Gi0/0/1 vlan 0 Unauthorized packet
*Mar 15 03:01:50.464: SISF[SWI]: Gi0/0/1 vlan 0 Feature Snooping rc 1
*Mar 15 03:01:50.464: SISF[SWI]: Gi0/0/1 vlan 0 Feature drop
*Mar 15 03:01:50.464: SISF[MEM]: Unlocking, count is now 0
*Mar 15 03:01:50.464: SISF[MEM]:  3BB56338 semaphore system unlocked
*Mar 15 03:01:54.548: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:01:57.543: SISF[SWI]: SISF IPv6 enqueue FE80::B614:89FF:FE03:2600
*Mar 15 03:01:59.428: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:02:03.896: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:02:08.500: SISF[SWI]: SISF IPv6 enqueue FE80::219:2FFF:FE53:83CE
*Mar 15 03:02:10.266: SISF[SWI]: SISF IPv6 enqueue FE80::213:80FF:FE3E:8B25
ASR1006-1#

Device# end
```

Related Commands

Command	Description
show mdns cache	Displays information about the resource records in the mDNS cache during the mDNS service discovery process.
show mdns requests	Displays information about the browse requests, pending service requests, and pending host resolve requests during the mDNS service discovery process.
show mdns statistics	Displays information about the number of packets sent, received, and dropped in the device during the mDNS service discovery process.

debug mdss

To display the run-time errors and sequence of events for the multicast distributed switching services (MDSS), use the **debug mdss**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mdss commanddebug mdss {**all** | **error** | **event**}
no debug mdss {**all** | **error** | **event**}

Syntax Description

all	Displays both errors and sequence of events for MDSS.
error	Displays the run-time errors for MDSS.
event	Displays the run-time sequence of events for MDSS.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows output using the **debug mdss**command with the **all** keyword:

```
Router# debug mdss all
mdss all debugging is on
Router# clear ip mroute *
Router#
01:31:03: MDSS: got MDFS_CLEARALL
01:31:03: MDSS: --> mdss_flush_all_sc
01:31:03: MDSS: enqueue a FE_GLOBAL_DELETE
01:31:03: MDSS: got MDFS_MROUTE_ADD for (0.0.0.0, 224.0.1.40)
01:31:03: MDSS: --> mdss_free_scmdb_cache
01:31:03: MDSS: got MDFS_MROUTE_ADD for (0.0.0.0, 239.255.158.197)
01:31:03: MDSS: got MDFS_MROUTE_ADD for (192.1.21.6, 239.255.158.197)
01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan22
01:31:03: MDSS: -- mdss_add_oif
01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan22
01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags |
MCACHE_MTU
01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan23
01:31:03: MDSS: -- mdss_add_oif
01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan23
01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags |
MCACHE_MTU
01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,
```

```
Vlan21) +Vlan24
01:31:03: MDSS: -- mdss_add_oif
01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan24
01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags |
MCACHE_MTU
01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan25
01:31:03: MDSS: -- mdss_add_oif
01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan25
01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags |
MCACHE_MTU
01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan26
01:31:03: MDSS: -- mdss_add_oif
01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197,
Vlan21) +Vlan26
01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags |
MCACHE_MTU
01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,u
Vlan21) +Vlan27
```

Related Commands

Command	Description
debug mls rp ip multicast	Displays information about MLSP.

debug media resource provisioning all

To display debugging messages related to all media resource provisioning, use the **debug media resource provisioning all** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug media resource provisioning all
no debug media resource provisioning all

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.

Examples

The following is sample output from the **debug media resource provisioning all** command:

```
Router# debug media resource provisioning all

.
.
.
Media resource provisioning all debugging is on.
Disabling profile will disconnect active CONFERENCING calls,
do you want to continue ? [yes/no]
*Jul  8 18:46:36: rpm_if_profile_exist ::profile id 10, service  TRANSCODING
*Jul  8 18:46:36: rpm_get_rscid_profile_info Profile with profile id :10, service :TRANSCODING
 does not exist
*Jul  8 18:46:36: rpm_if_profile_exist ::profile id 10, service  CONFERENCING
*Jul  8 18:46:36: rpm_if_profile_exist ::profile id 10, service  TRANSCODING
*Jul  8 18:46:36: rpm_get_rscid_profile_info Profile with profile id :10, service :TRANSCODING
 does not exist
*Jul  8 18:46:36: rpm_if_profile_exist ::profile id 10, service  CONFERENCING
 Must be yes or no
Router(config-dspfarm-profile)#
Router(config-dspfarm-profile)#
Router(config-dspfarm-profile)#
Router(config-dspfarm-profile)# no shutdown

Router(config-dspfarm-profile)#
*Jul  8 18:46:42: rpm_user_enable_profile ::profile id 10, service CONFERENCING
*Jul  8 18:46:44:%DSPRM-5-UPDOWN:DSP 10 in slot 1, changed state to up
*Jul  8 18:46:44: rpm_rscprv_update ::provider_id 1 rsc_id 2 rsc_grp_state 4num_channel_delta
 0
*Jul  8 18:46:44: rpm_rscprv_update resource update from resource provider 1 is successful
Router(config-dspfarm-profile)#
Router(config-dspfarm-profile)# exit

Router(config)# exit
```

Related Commands

Command	Description
debug media resource provisioning errors	Displays debugging messages related to media resource provisioning errors.
debug media resource provisioning events	Displays debugging messages related to media resource provisioning events.

debug media resource provisioning errors

To display debugging messages related to media resource provisioning errors, use the **debug media resource provisioning errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug media resource provisioning errors
no debug media resource provisioning errors

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following is sample output from the **debug media resource provisioning errors** command:

```
Router# debug media resource provisioning errors

Media resource provisioning errors debugging is on
Router# no debug media resource provisioning errors

Media resource provisioning errors debugging is off
```

Related Commands

Command	Description
debug media resource provisioning all	Displays debugging messages related to all media resource provisioning.
debug media resource provisioning events	Displays debugging messages related to media resource provisioning events.

debug media resource provisioning events

To display debugging messages related to media resource provisioning events, use the **debug media resource provisioning events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug media resource provisioning events
no debug media resource provisioning events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following is sample output from the **debug media resource provisioning events** command:

```
Router# debug media resource provisioning events

Media resource provisioning events debugging is on
Router# no debug media resource provisioning events

Media resource provisioning events debugging is off
Router#
```

Related Commands

Command	Description
debug media resource provisioning all	Displays debugging messages related to all media resource provisioning.
debug media resource provisioning errors	Displays debugging messages related to media resource provisioning errors.

debug mediacard

To display Digital Signal Processor Resource Manager (DSPRM) debugging information, use the **debug mediacard** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mediacard {**all** | **errors** | **events** | **message**}
no debug mediacard {**all** | **errors** | **events** | **message**}

Syntax Description

all	Debugs DSPRM errors, events, and messages.
errors	Debugs DSPRM errors.
events	Debugs DSPRM events.
message	Debugs DSPRM messages.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)XY	This command was introduced on the Communication Media Module.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.
12.4(3)	This command was integrated into Cisco IOS Release 12.4(3).

Usage Guidelines

Use the **debug mediacard errors** command to debug active calls. You should use the **debug mediacard all** command during minimum traffic periods only; using the **debug mediacard all** command during active calls can significantly impact system performance.

Examples

The following is sample output from the **debug mediacard** command:

```
Router# debug mediacard messages
Media Card service messages debugging is on
*Mar  1 07:45:06.362: >  CREATE_CONFERENCE (0x1) , pktLen 56, confId 1, instId 1
7483, seqNo 27983, Payload (24 bytes): confType 3, agcMode 1, spkrUpdateReportMo
de 1, maxActSpkr 3
*Mar  1 07:45:06.362: >  CREATE_CHANNEL (0x64) , pktLen 100, confId 1, instId 26
625, seqNo 27984, Payload (68 bytes): rxCodecType 1, suppressRx 1, rxCNG 2, rxPL
C 2, rxVAD 2, rxToneDet 1, rxSpkrPriority 1, rxInactiveTimeOut 7200, rxPacketSiz
e 20, rxRTPPayloadType 0
*Mar  1 07:45:06.362:    txCodecType 2, suppressTx 1, txVAD 1, AGC 1, txSSRC 167
860472, txPacketSize 20, txRTPPayloadType 0
*Mar  1 07:45:06.362:  < CREATE_CONFERENCE_ACK (0x4001) , pktLen 116, confId 1,
instId 0, seqNo 27983, Payload (84 bytes): status 0 (Normal Completion), param1
3, param2 0
*Mar  1 07:45:06.362:  < CREATE_CHANNEL_ACK (0x4064) , pktLen 116, confId 1, ins
tId 26625, seqNo 27984, Payload (84 bytes): status 0 (Normal Completion), param1
 0, param2 0
```

```
*Mar  1 07:45:06.362: >  CREATE_CONFERENCE (0x1) , pktLen 56, confId 2, instId
All possible debugging has been turned off
MTP#26625, seqNo 27985, Payload (24 bytes): confType 3, agcMode 1, spkrUpdateRep
ortMode 1, maxActSpkr 3
*Mar  1 07:45:06.362: >  CREATE_CHANNEL (0x64) , pktLen 100, confId 2, instId 26
626, seqNo 27986, Payload (68 bytes): rxCodecType 2, suppressRx 1, rxCNG 2, rxPL
C 2, rxVAD 2, rxToneDet 1, rxSpkrPriority 1, rxInactiveTimeOut 7200, rxPacketSiz
e 20, rxRTPPayloadType 0
*Mar  1 07:45:06.366:    txCodecType 1, suppressTx 1, txVAD 1, AGC 1, txSSRC 167
858296, txPacketSize 20, txRTPPayloadType 0
*Mar  1 07:45:06.366:  < CREATE_CONFERENCE_ACK (0x4001) , pktLen 116, confId 2,
instId 0, seqNo 27985, Payload (84 bytes): status 0 (Normal Completion), param1
3, param2 0
Router# debug mediacard events
Media Card service events debugging is on
*Mar  1 07:47:53.926: ms_ac_open_rtp_sockets: loc_ipaddr = 10.1.80.24 loc_mac<00
03.feac.c842> rem_ip<0.0.0.0> rem_port<0>
*Mar  1 07:47:53.926: ms_ac_get_unique_udp_port: rtcp_socket = 6255F490
*Mar  1 07:47:53.926: ms_ac_get_unique_udp_port: SLOT3 Port<3450> is assigned!
*Mar  1 07:47:53.926: ms_ac_open_local_rtp: rtpinfo 64382A3C, local_port =23930
*Mar  1 07:47:53.926: ms_ac_rtp_enq: Sent msg 101 to DSPFARM
*Mar  1 07:47:53.926: ms_ac_open_remote_rtp: rtpinfo 64382A3C, loc_ipaddr = 10.1
.80.24 loc_udp_prt <23930> ,loc_mac<0003.feac.c842>
*Mar  1 07:47:53.926: ms_ac_open_remote_rtp: remote_ipaddr = 10.1.2.15 remote_ud
p_prt <17932>
*Mar  1 07:47:53.926: ms_ac_nexthop_macaddr idb<630BDFCC> nexthop<10.1.80.1>
*Mar  1 07:47:53.926: ms_ac_nexthop_macaddr ptr<6301F5AC> through<GigabitEtherne
t1/0> nexthop<10.1.80.1>
*Mar  1 07:47:53.926: ms_ac_after_found_mac <10.1.2.15>'s mac <00d0.002a.7400> f
ound
*Mar  1 07:47:53.926: ms_ac_check_xcode_rem_ip: rtpinfo <64382A3C> other_rtpinfo
<0>
*Mar  1 07:47:53.926: ms_ac_rtp_enq: Sent msg 103 to DSPFARM
*Mar  1 07:47:53.942: ms_ac_open_rtp_sockets: loc_ipaddr = 10.1.80.24 loc_mac<00
03.feac.c842> rem_ip<0.0.0.0> rem_port<0>
*Mar  1 07:47:53.942: ms_ac_get_unique_udp_port: rtcp_socket = 6256C9B4
*Mar  1 07:47:53.942: ms_ac_get_unique_udp_port: SLOT3 Port<1778> is assigned!
*Mar  1 07:47:53.942: ms_ac_open_local_rtp: rtpinfo 6438353C, local_port =22258
*Mar  1 07:47:53.942: ms_ac_rtp_enq: Sent msg 101 to DSPFARM
*Mar  1 07:47:53.942: ac_validate_xcode_params: codeDec<2> codeEnc<1> decDur<20>
 encDur<20>
*Mar  1 07:47:53.942: ac_open_xcode_channel: codeDec<1> codeEnc<2> decDur<20> en
cDur<20> VADen<0> prf_id<4>
*Mar  1 07:47:53.942: reserve_xcode_resource: reserve xcode resource:codecDec<1>
 codecEnc<2>
*Mar  1 07:47:53.942: al
```

Related Commands

Command	Description
show mediacard	Displays information about the media card.

debug memory

To enable debugging on memory, use the **debug memory**command in privileged EXEC mode. To disable memory debugging, use the **no** form of this command.

debug memory [**rmi**]
no debug memory

Syntax Description

rmi	(Optional) Displays debug information related to memory Remote Method Invocation (RMI).

Command Default

Memory debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(2)T	This command was introduced.

Usage Guidelines

The **debug memory** command is used when debugging memory manager operations such as allocating and reallocating memory.

Examples

The following example shows how to enable memory debugging:

```
Router# debug memory
Memory debugging is on
```

The following example shows how to enable memory RMI debugging:

```
Router# debug memory rmi
Memory RMI debugging is on
```

Related Commands

Command	Description
show debug	Displays the types of debugging that are enabled.

debug metadata

To enable debugging for metadata flow information, use the **debug metadata** command in privileged EXEC mode. To disable debugging for metadata flow information, use the **no** form of this command.

debug metadata {**encode-decode** {**details** | **errors** | **events**} | **flow** {**all** | **core** | **table**} | **nbar**}
no debug metadata {**encode-decode** {**details** | **errors** | **events**} | **flow** {**all** | **core** | **table**} | **nbar**}

Syntax Description

encode-decode	Debugs information related to the metadata encoding and decoding mechanism.
details	Debugs details that occurred during the encode-decode process.
errors	Debugs errors that occurred during the encode-decode process.
events	Debugs events that occurred during the encode-decode process.
flow	Debugs details related to metadata flow.
all	Debugs all metadata flow information.
core	Debugs core metadata events information.
table	Debugs metadata flow table information.
nbar	Debugs Network-Based Application Recognition (NBAR) as a source for metadata.

Command Default

Debugging for metadata flow information is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.2(1)T	This command was introduced.
15.2(4)M	This command was modified. The **nbar** keyword was added.

Examples

The following is sample output from the **debug metadata encode-decode details** command. The debug output shows the process for creating the IP information export (IPFIX) template and decoding the metadata information. The last two lines indicate the length, Variable Length Information ID (VLIE), and metadata application name.

```
Device# debug metadata encode-decode details

*Jul 14 03:24:50.395: MED-IPFIX: Hdr: Ver 10 msg len 66
*Jul 14 03:24:50.395: MED-IPFIX: Hdr: Export time = Thu
Jul 14 08:54:50 2011
*Jul 14 03:24:50.395: MED-IPFIX: Hdr: Seq num = 4
*Jul 14 03:24:50.395: MED-IPFIX: Hdr: Obs dom ID = 0
*Jul 14 03:24:50.395: MED-IPFIX: Creating IP FIX Template, 79CD778
```

```
*Jul 14 03:24:50.395: MED-IPFIX: Decoded and saved ID 256 Templates Address 79CD778
*Jul 14 03:24:50.395: MED-IPFIX: Decoding 2 Template fields
*Jul 14 03:24:50.395: MED-IPFIX: len=4 936750775487430656
*Jul 14 03:24:50.395: MED-IPFIX: VLIE len 17 [telepresence-data]
```

The following is sample output from the **debug metadata flow all** command. The first few lines in the output display the addition of an event. Then, the output shows details of ingress and egress interfaces. Next, the display shows various application names and the associated application IDs. Then, Classification types and the matching applications follow.

The last line, "DB Addition Succeded" indicates that an appropriate match was detected and the control plane classification completed successfully.

```
Device# debug metadata flow all

*Jul 14 08:07:23.155: FMD SIG: Process RSVP Event RSVP_FMD_EVENT_PAYLOAD_RECEIVED(1)
*Jul 14 08:07:23.155: FMD : fmd_post_events: posting event 0
*Jul 14 08:07:23.167: FMD Process Event - FMD_RSVP_TRANSPORT_ADD
*Jul 14 08:07:23.167: (fmd_add_event_process): For Source IP/Port : 67372036/1000
*Jul 14 08:07:23.167: FMD DB Lookup: Hash 391
*Jul 14 08:07:23.167: FMD Event for Ingress Interface Ethernet0/0 , Egress Interface
Ethernet0/1
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 17, Value telepresence-data
*Jul 14 08:07:23.167: FMD Classification Dest Type 95, Len 4, Value
*Jul 14 08:07:23.167: App name telepresence-data id 218104286 in Metadata local app table
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 11, Value webex-audio
*Jul 14 08:07:23.167: FMD Classification Dest Type 95, Len 4, Value
*Jul 14 08:07:23.167: App name webex-audio id 12 in Metadata local app table
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 11, Value webex-audio
*Jul 14 08:07:23.167: FMD Classification Dest Type 96, Len 17, Value telepresence-data *Jul
 14 08:07:23.167: FMD Classification Src Type 96, Len 11, Value webex-audio
*Jul 14 08:07:23.167: FMD Classification Dest Type 0, Len 0, Value
*Jul 14 08:07:23.167: FMD Classification: Match Passed for type 95 value Router-201
*Jul 14 08:07:23.167: FMD Classification: Found 1 filters matching
*Jul 14 08:07:23.167: FMD Event: Input policy Matched, Add flow to CFT
*Jul 14 08:07:23.167: FMD Event: PPCP Binding Succeeded
*Jul 14 08:07:23.167: FMD fmd_add_update_ingress_cft_fo : fid 4
*Jul 14 08:07:23.167: FMD Event: Local Flow ID 0
*Jul 14 08:07:23.167: (fmd_add_event_process): Update with Template Address 79CD778, Md
Addr 947F810
*Jul 14 08:07:23.167: fmd_add_ipv4_flow_node_to_hash: Hash 391
*Jul 14 08:07:23.167: FMD Event: DB Addition Succeeded
```

The following is sample output from the **debug metadata nbar** command. The fields are self-explanatory.

```
Device# debug metadata nbar

*May 21 10:22:02.655: FMD NBAR: Successfully activated NBAR for proto id: 64
*May 21 10:22:02.656: FMD NBAR: fmd filter "application telepresence-media"
*May 21 10:22:02.656: FMD NBAR: Match application command found
*May 21 10:22:02.656: FMD NBAR: Successfully activated NBAR for proto id: 113
*May 21 10:22:02.656: FMD NBAR: class_id 0 name class-default
*May 21 10:22:02.656: FMD NBAR: Non Metadata filter type 26. Skipping
```

Related Commands

Command	Description
metadata application-params	Creates new metadata application parameters.
show metadata application table	Displays a list of metadata applications defined on a device.

Command	Description
show metadata flow	Displays the metadata flow information.

debug mgcp

To enable debug traces for Media Gateway Control Protocol (MGCP) errors, events, media, packets, parser, and Call Admission Control (CAC), use the **debug mgcp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp [{**all** | **errors** [**endpoint** *endpoint-name*] | **events** [**endpoint** *endpoint-name*] | **media** [**endpoint** *endpoint-name*] | **nas** | **packets** [{**endpoint** *endpoint-name* | **input-hex**}] | **parser** | **src** | **voipcac**}]
no debug mgcp [{**all** | **errors** | **events** | **media** | **nas** | **packets** | **parser** | **src** | **voipcac**}]

Syntax Description

all	(Optional) Debugs MGCP errors, events, media, packets, parser and builder, and CAC.
errors	(Optional) Debugs MGCP errors.
endpoint *endpoint-name*	(Optional) Debugs MGCP errors, events, media, or packets per endpoint.
events	(Optional) Debugs MGCP events.
media	(Optional) Debugs MGCP tone and signal events.
nas	(Optional) Debugs MGCP network access server (NAS) (data) events.
packets	(Optional) Debugs MGCP packets.
input-hex	(Optional) Debugs MGCP input packets in hexadecimal values.
parser	(Optional) Debugs MGCP parser and builder.
src	(Optional) Debugs MGCP System Resource Check (SRC) CAC information.
voipcac	(Optional) Turns on debugging messages for the Voice over IP (VoIP) CAC process at the MGCP application layer.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1(3)T	Additional information was displayed for the gateways.
12.1(5)XM, 12.2(2)T	The output was modified to display parameters for the MGCP channel-associated signaling (CAS) PBX and ATM adaptation layer 2 (AAL2) permanent virtual circuit (PVC) features.
12.2(2)XA	The **media** keyword was added. The **endpoint** *endpoint-name*keyword and argument were added as options for the **errors**, **events**, **media,** and **packets** keywords. The **input-hex** keyword option was added for the **packets**keyword.

Release	Modification
12.2(2)XB	The **nas** keyword and the **src** and **voipcac** keywords were added. (Refer to MGCP VoIP Call Admission Control in Cisco IOS Release 12.2(2)XB.)
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T. **Note** The **nas** keyword was not integrated into Cisco IOS Release 12.2(8)T.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.

Usage Guidelines

There is always a performance penalty when using debug commands.

Examples

The following is sample output from the **debug mgcp errors**, **debug mgcp events**, **debug mgcp media**, **debug mgcp nas**, **debug mgcp packets**, **debug mgcp parser**, and **debug mgcp src** commands and keywords. The **debug mgcp all** command and keyword would show a compilation of all this output, including the **debug mgcp voipcac** command and keyword output. Note that using the **debug mgcp all** command and keyword may severely impact network performance.

The following is sample output from the **debug mgcp errors** command and keyword:

```
Router# debug mgcp errors
Unknown network interface type
```

The following is sample output from the **debug mgcp events** command and keyword:

```
Router# debug mgcp events
Media Gateway Control Protocol events debugging is on
Router#
1w1d: MGC stat - 172.19.184.65, total=44, succ=7, failed=21
1w1d: MGCP msg 1
1w1d: remove_old_under_specified_ack:
1w1d: MGC stat - 172.19.184.65, total=44, succ=8, failed=21
1w1d: updating lport with 2427setup_ipsocket: laddr=172.29.248.193, lport=2427,
faddr=172.19.184.65, fport=2427
1w1d: enqueue_ack: ackqhead=0, ackqtail=0, ackp=1DC1D38, msg=21A037C
```

The following is sample output from the **debug mgcp media** command and keyword:

```
Router# debug mgcp media
Media Gateway Control Protocol media events debugging is on
Router#
DYNAMIC payload type
DYNAMIC payload type
*Jan  1 03:02:13.159:mgcp_verify_supp_reqdet_ev
*Jan  1 03:02:13.159:mgcp_verify_supp_signal_ev
*Jan  1 03:02:13.159:process_request_ev- callp 635368FC, voice_if 6353C1F8
*Jan  1 03:02:13.159:process_detect_ev- callp 635368FC, voice_if 6353C1F8
*Jan  1 03:02:13.159:process_signal_ev- callp 635368FC, voice_ifp 6353C1F8
*Jan  1 03:02:13.159:mgcp_process_quarantine_mode- callp 635368FC, voice_if 6353C1F8
*Jan  1 03:02:13.159:mgcp_process_quarantine_mode- new q mode:process=0, loop=0
*Jan  1 03:02:13.179:process_deferred_request_events
*Jan  1 03:02:13.479:mgcp_verify_supp_reqdet_ev
*Jan  1 03:02:13.479:mgcp_verify_supp_signal_ev
```

```
*Jan  1 03:02:13.479:process_request_ev- callp 6353BCCC, voice_if 638C3094
*Jan  1 03:02:13.479:process_detect_ev- callp 6353BCCC, voice_if 638C3094
*Jan  1 03:02:13.479:process_signal_ev- callp 6353BCCC, voice_ifp 638C3094
*Jan  1 03:02:13.479:mgcp_process_quarantine_mode- callp 6353BCCC, voice_if 638C3094
*Jan  1 03:02:13.479:mgcp_process_quarantine_mode- new q mode:process=0, loop=0
*Jan  1 03:02:13.499:process_deferred_request_events
*Jan  1 03:02:13.827:mgcp_verify_supp_reqdet_ev
*Jan  1 03:02:13.827:mgcp_verify_supp_signal_ev
*Jan  1 03:02:13.827:process_request_ev- callp 635368FC, voice_if 6353C1F8
*Jan  1 03:02:13.827:process_detect_ev- callp 635368FC, voice_if 6353C1F8
*Jan  1 03:02:13.827:process_signal_ev- callp 635368FC, voice_ifp 6353C1F8
*Jan  1 03:02:13.827:mgcp_process_quarantine_mode- callp 635368FC, voice_if 6353C1F8
*Jan  1 03:02:13.827:mgcp_process_quarantine_mode- new q mode:process=0, loop=0
*Jan  1 03:02:13.831:process_deferred_request_events
*Jan  1 03:02:23.163:mgcp_cr_and_init_evt_node:$$$ the node pointer 63520B14
*Jan  1 03:02:23.163:mgcp_insert_node_to_preprocess_q:$$$enq to preprocess, qhead=63520B14,
 qtail=63520B14, count 1, evtptr=63520B14
*Jan  1 03:02:23.479:mgcp_cr_and_init_evt_node:$$$ the node pointer 63520BA8
*Jan  1 03:02:23.479:mgcp_insert_node_to_preprocess_q:$$$enq to preprocess, qhead=63520BA8,
 qtail=63520BA8, count 1, evtptr=63520BA8
```

The following is sample output for the **debug mgcp nas** command and keyword, with the **debug mgcp packets** command and keyword enabled as well:

```
Router# debug mgcp nas
Media Gateway Control Protocol nas pkg events debugging is on
Router# debug mgcp packets
Media Gateway Control Protocol packets debugging is on
Router#
01:49:14:MGCP Packet received -
CRCX 58 S7/DS1-0/23 MGCP 1.0
X:57
 M:nas/data
 C:3

 L:b:64, nas/bt:modem, nas/cdn:3000, nas/cgn:1000

mgcp_parse_conn_mode :string past nas = data
mgcp_chq_nas_pkg:Full string:nas/bt:modem
mgcp_chq_nas_pkg:string past slash:bt
mgcp_chq_nas_pkg:string past colon:modem
mgcp_chq_nas_pkg:Full string:nas/cdn:3000
mgcp_chq_nas_pkg:string past slash:cdn
mgcp_chq_nas_pkg:string past colon:3000
mgcp_chq_nas_pkg:Full string:nas/cgn:1000
c5400#
mgcp_chq_nas_pkg:string past slash:cgn
mgcp_chq_nas_pkg:string past colon:1000
CHECK DATA CALL for S7/DS1-0/23
 mgcpapp_xcsp_get_chan_cb -Found - Channel state  Idle
 CRCX Recv
mgcpapp_endpt_is_data:endpt S7/DS1-0/23, slot 7, port 0 chan 23
 mgcpapp_data_call_hnd:mgcpapp_xcsp_get_chan_cb -Found - Channel state  Idle
bw=64, bearer=E1,cdn=3000,cgn=1000
```

The following is sample output from the **debug mgcp packets** command and keyword:

```
Router# debug mgcp packets
Media Gateway Control Protocol packets debugging is on
Router#
1w1d: MGCP Packet received -
DLCX 408631346 * MGCP 0.1
1w1d: send_mgcp_msg, MGCP Packet sent --->
```

```
1w1d: 250 408631346
<---
```

The following is sample output from the **debug mgcp parser** command and keyword:

```
Router# debug mgcp

parser
Media Gateway Control Protocol parser debugging is on
Router#
1w1d: -- mgcp_parse_packet() - call mgcp_parse_header
- mgcp_parse_header()- Request Verb FOUND DLCX
- mgcp_parse_packet() - out mgcp_parse_header
- SUCCESS: mgcp_parse_packet()- MGCP Header parsing was OK
- mgcp_val_mandatory_parms()
- SUCCESS: mgcp_parse_packet()- END of Parsing
1w1d: -- mgcp_build_packet()-
1w1d: - mgcp_estimate_msg_buf_length() - 87 bytes needed for header
- mgcp_estimate_msg_buf_length() - 87 bytes needed after checking parameter lines
- mgcp_estimate_msg_buf_length() - 87 bytes needed after checking SDP lines
- SUCCESS: MGCP message building OK
- SUCCESS: END of building
```

The following is sample output from the **debug mgcp src** command and keyword:

```
Router# debug mgcp src
Media Gateway Control Protocol System Resource Check CAC debugging is on
Router#
00:14:08: setup_indication: Set incoming_call flag=TRUE in voice_if
00:14:08: send_mgcp_msg, MGCP Packet sent --->
00:14:08: NTFY 11 aaln/S1/1@Router MGCP 0.1
N: emu@[1.4.173.1]:51665
X: 35
O: hd
<---
00:14:08: MGCP Packet received -
200 11 hello
00:14:08: MGCP Packet received -
RQNT 42 aaln/S1/1 MGCP 0.1
N: emu@[1.4.173.1]:51665
X: 41
R: D/[0-9*#T](d), hu
S: dl
D: (911|xxxx)
00:14:08: send_mgcp_msg, MGCP Packet sent --->
00:14:08: 200 42 OK
<---
00:14:12: send_mgcp_msg, MGCP Packet sent --->
00:14:12: NTFY 12 aaln/S1/1@Router MGCP 0.1
N: emu@[1.4.173.1]:51665
X: 41
O: D/2222
<---
00:14:12: MGCP Packet received -
200 12 phone-number ok
00:14:12: MGCP Packet received -
CRCX 44 aaln/S1/1 MGCP 0.1
N: emu@[1.4.173.1]:51665
C: 3
X: 43
R: hu(n)
M: recvonly
L: a:G.711u,p:5,e:off,s:off
```

```
00:14:12: mgcp_setup_conn_check_system_resource: System resource check successful
00:14:12: mgcp_voice_crcx: System resource is available
00:14:12: mgcp_set_call_counter_control: Incoming call with 1 network leg, flag=FALSE
00:14:12: send_mgcp_msg, MGCP Packet sent --->
00:14:12: 200 44
I: 4
v=0
o=- 4 0 IN IP4 1.4.120.1
s=Cisco SDP 0
c=IN IP4 1.4.120.1
t=0 0
m=audio 16404 RTP/AVP 0
<---
00:14:13: MGCP Packet received -
MDCX 48 aaln/S1/1 MGCP 0.1
N: emu@[1.4.173.1]:51665
C: 3
I: 4
X: 47
M: recvonly
R: hu
L: a:G.711u,p:5,e:off,s:off
v=0
o=- 4 0 IN IP4 1.4.120.3
s=Cisco SDP 0
c=IN IP4 1.4.120.3
t=0 0
m=audio 16384 RTP/AVP 0
00:14:13: mgcp_modify_conn_check_system_resource: System resource check successful
00:14:13: mgcp_modify_connection: System resource is available
00:14:13: send_mgcp_msg, MGCP Packet sent --->
00:14:13: 200 48 OK
<---
00:14:20: MGCP Packet received -
MDCX 52 aaln/S1/1 MGCP 0.1
N: emu@[1.4.173.1]:51665
C: 3
I: 4
X: 51
M: sendrecv
R: hu
L: a:G.711u,p:5,e:off,s:off
00:14:20: mgcp_modify_conn_check_system_resource: System resource check successful
00:14:20: mgcp_modify_connection: System resource is available
00:14:20: send_mgcp_msg, MGCP Packet sent --->
00:14:20: 200 52 OK
<---
00:14:34: MGCP Packet received -
DLCX 56 aaln/S1/1 MGCP 0.1
X: 55
N: emu@[1.4.173.1]:51665
C: 3
I: 4
R: hu
00:14:34: send_mgcp_msg, MGCP Packet sent --->
00:14:34: 250 56
P: PS=1382, OS=110180, PR=1378, OR=109936, PL=63484, JI=520, LA=2
<---
00:14:36: mgcp_reset_call_direction: Reseting incoming_call flag=FALSE in voice_if
00:14:36: send_mgcp_msg, MGCP Packet sent --->
00:14:36: NTFY 13 aaln/S1/1@tlkrgw1 MGCP 0.1
N: emu@[1.4.173.1]:51665
X: 55
```

```
O: hu
<---
```

debug mgcp all

To enable all debug traces for Media Gateway Control Protocol (MGCP), use the **debug mgcp all**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp all [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp all

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1(3)T	Additional information was displayed for the gateways.
12.1(5)XM, 12.2(2)T	The output was modified to display parameters for the MGCP channel-associated signaling (CAS) PBX and ATM adaptation layer 2 (AAL2) permanent virtual circuit (PVC) features.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

This command enables the following MGCP debug commands:

- **debug mgcp endptdb**
- **debug mgcp errors**
- **debug mgcp events**
- **debug mgcp gcfm**
- **debug mgcp inout**
- **debug mgcp media**

- **debug mgcp nas**
- **debug mgcp packets**
- **debug mgcp parser**
- **debug mgcp src**
- **debug mgcp state**
- **debug mgcp voipcac**

Caution Using the **debug mgcp all** command may severely impact network performance.

Examples

The following is sample output from the **debug mgcp all**command:

```
Router# debug mgcp all
This may severely impact network performance. Continue[confirm]
Media Gateway Control Protocol all debugging is on, trace-level Verbose
Router#
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_count_active_mgc_msg_stat(240):[lvl=1]MGC
 stat - 192.168.1.200, total=8, succ=5, failed=1
*Sep 10 17:20:24.408: MGCP Packet received from 192.168.1.200:7979--->
CRCX 6 aaln/S2/SU1/1 MGCP 1.0
M: recvonly
C: 1
<---
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcpapp_process_mgcp_msg(3318):[lvl=0] : <NEW
 MGCP MSG From CA>
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(316):[lvl=0]call
mgcp_parse_header
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(320):[lvl=0]out
mgcp_parse_header
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(360):[lvl=1]SUCCESS: - MGCP
 Header parsing was OK
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_string_parse(186):[lvl=0]return code=1.
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_parameter_lines(725):[lvl=1]return
 parse function in mgcp_parm_rules_array[6].
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4762):[lvl=0](in_ptr:
recvonly)
*Sep 10 17:20:24.408:
//-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4780):[lvl=0]tmp_ptr:(recvonly)
*Sep 10 17:20:24.408:
//-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4816):[lvl=0]tmp_ptr:(recvonly)
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4822):[lvl=0]match recvonly
 recvonly
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4830):[lvl=0]case
MODE_RECVONLY
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4894):[lvl=0]SUCCESS:
Connection Mode parsing is OK
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_string_parse(186):[lvl=0]return code=1.
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_parameter_lines(725):[lvl=1]return
 parse function in mgcp_parm_rules_array[1].
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_call_id(840):[lvl=0]in_ptr: 1
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_call_id(883):[lvl=1]SUCCESS: Call
ID string(1) parsing is OK
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_mandatory_parms(12428):[lvl=0]Entered
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_comp_mp_parms(14923):[lvl=0]Entered
```

```
*Sep 10 17:20:24.408: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_comp_mp_parms(14928):[lvl=1] -
lcon_opt_ptr could not be obtained
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(378):[lvl=2]SUCCESS: END of
 Parsing
*Sep 10 17:20:24.412:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_a(1339):[lvl=0]aaln/S2/SU1/1
*Sep 10 17:20:24.412:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_slot(1632):[lvl=0]2/SU1/1
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]2/SU1/1
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_slot(1641):[lvl=0]
 : ifn 0x665449A8, slot:2
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_su(1773):[lvl=0]1/1
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]1/1
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_port(1807):[lvl=0]1
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]1
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_endpt_offset(2590):[lvl=0]endpt
 NULL
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_by_ifn(1326):[lvl=0]Entered
*Sep 10 17:20:24.412:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_tree_link_by_ifn(1145):[lvl=0]Entered
*Sep 10 17:20:24.412: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_compute_key(196):[lvl=0]type 2 slot
 0002 subunit 0001
*Sep 10 17:20:24.412:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_tree_link_by_ifn(1157):[lvl=0]computed key 0x2081FF01
```

Related Commands

Command	Description
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show debugging	Displays the types of debugging that are enabled.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp endpoint

To enable debug traces for a specific Media Gateway Control Protocol (MGCP) endpoint, use the **debug mgcp endpoint** command in privileged EXEC mode. To disable debugging output for the endpoint, use the **no** form of this command.

debug mgcp endpoint *endpoint-name* {**all** [**tracelevel** {**critical** | **moderate** | **verbose**}] | **errors** | **events** [**tracelevel** {**critical** | **moderate** | **verbose**}] | **media** [**tracelevel** {**critical** | **moderate** | **verbose**}] | **packets**}
no debug mgcp endpoint *endpoint-name* {**all** | **errors** | **events** | **media** | **packets**}

Syntax Description

endpoint-name	Name of the MGCP endpoint for which to enable debugging. Must be a fully specified and supported endpoint.
all	Displays MGCP errors, events, media, and packets for the specified endpoint.
errors	Displays MGCP errors for the specified endpoint.
events	Displays MGCP events for the specified endpoint.
media	Displays MGCP tone and signal events for the specified endpoint.
packets	Displays MGCP packets for the specified endpoint.
tracelevel	(Optional) Sets the priority level for the **all**, **events**, or **media** debug trace. • **critical** --Displays only high-priority debug information. • **moderate** --Displays medium and high-priority debug information. • **verbose** --Displays all debug information. This is the default level. **Note** This keyword is not available for **errors** or **packets** debugging.

Command Default

Debugging for specific endpoints is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(4)T	This command was introduced.

Usage Guidelines

This command enables debugging for a specific MGCP endpoint. You can enable the same type of debugging globally for all endpoints by using the **debug mgcp all**, **debug mgcp errors**, **debug mgcp events**, **debug mgcp media**, or **debug mgcp packets** commands.

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

This command sets the trace level for the specific endpoint. You can set the trace level globally for all MGCP debug commands and endpoints by using the **debug mgcp tracelevel-default** command. Setting the endpoint-specific trace level takes precedence over the global trace-level.

Note Trace levels are not supported for errors or packets debugging because all of the output from those commands is set to high priority.

Examples

The following is sample output from the **debug mgcp endpoint**command:

```
Router# debug mgcp endpoint aaln/S2/SU1/1 events tracelevel critical
Media Gateway Control Protocol events debugging for endpoint aaln/S2/SU1/1 is on, trace-level
 Critical
Router#
*Sep 10 17:46:13.100:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_idle_crcx(4875):[lvl=2]callp(0x63E313E0),
current state CALL_IDLE, event EV_CREATE_CONN
*Sep 10 17:46:13.100:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E313E0), new state CALL_CONNECTING, event EV_CREATE_CONN
*Sep 10 17:46:13.104:
//8/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_call_pre_conference(223):[lvl=2]callp(0x63E311D0),
 current state CALL_CONNECTING, event EV_CALL_CONNECT
*Sep 10 17:46:13.104:
//8/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_call_connect(7331):[lvl=2]callp(0x63E311D0),
 current state CALL_CONNECTING, event EV_CALL_CONNECT
*Sep 10 17:46:13.104:
//8/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E311D0), new state CALL_CONFERENCING, event EV_CALL_CONNECT
*Sep 10 17:46:13.104:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_call_proceeding(6306):[lvl=2]callp(0x63E313E0),
 current state CALL_CONNECTING, event EV_CALL_PROCEED
*Sep 10 17:46:13.104:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_call_connect(7331):[lvl=2]callp(0x63E313E0),
 current state CALL_CONNECTING, event EV_CALL_PROCEED
*Sep 10 17:46:13.104:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E313E0), new state CALL_CONFERENCING, event EV_CALL_PROCEED
*Sep 10 17:46:13.108:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_ignore_ccapi_ev(4316):[lvl=2]callp(0x63E313E0),
 current state CALL_CONFERENCING, event EV_CONF_RDY
*Sep 10 17:46:13.108:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E313E0), new state CALL_CONFERENCING, event EV_CONF_RDY
*Sep 10 17:46:13.108:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_call_modified(7710):[lvl=2]callp(0x63E313E0),
 current state CALL_CONFERENCING, event EV_MODIFY_DONE
*Sep 10 17:46:13.108:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E313E0), new state CALL_CONFERENCING, event EV_MODIFY_DONE
*Sep 10 17:46:13.108:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_voice_mode_done(7994):[lvl=2]callp(0x63E313E0),
 current state CALL_CONFERENCING, event EV_VOICE_MODE_DONE, minor ev(d): 138, minor ev
*Sep 10 17:46:13.112:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E313E0), new state CALL_ACTIVE, event EV_VOICE_MODE_DONE
*Sep 10 17:46:23.104:
//7/9D04EB218005/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_invoke_app_sm(570):[lvl=2]MGCP:FSM
done- callp(63E313E0), new state CALL_ACTIVE, event EV_MEDIA_EVT
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp errors	Enables debug traces for MGCP errors.
debug mgcp events	Enables debug traces for MGCP events.
debug mgcp media	Enables debug traces for MGCP tone and signal events.
debug mgcp packets	Enables debug traces for MGCP packets.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show debugging	Displays the types of debugging that are enabled.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp endptdb

To enable debug traces for all Media Gateway Control Protocol (MGCP) endpoints, use the **debug mgcp endptdb** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp endptdb [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp endptdb

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP debugging for endpoints is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XA	This command was introduced.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

This command enables debugging globally for all MGCP endpoints. You can limit debugging to a specific endpoint by using the **debug mgcp endpoint** command.

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp endptdb** command used with the **debug mgcp packets**command:

```
Router# debug mgcp packets
Media Gateway Control Protocol packets debugging for all endpoints is on
Router# debug mgcp endptdb
Media Gateway Control Protocol endpoint database debugging for all endpoints is on,
trace-level Verbose
Router#
*Sep 10 11:39:16.467: MGCP Packet received from 192.168.1.200:7979--->
CRCX 27 aaln/S2/SU1/1 MGCP 1.0
M: recvonly
C: 1
<---
```

```
*Sep 10 11:39:16.467:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_a(1339):[lvl=0]aaln/S2/SU1/1
*Sep 10 11:39:16.467:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_slot(1632):[lvl=0]2/SU1/1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]2/SU1/1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_slot(1641):[lvl=0]
 : ifn 0x665449A8, slot:2
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_su(1773):[lvl=0]1/1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]1/1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_port(1807):[lvl=0]1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_by_ifn(1326):[lvl=0]Entered
*Sep 10 11:39:16.467:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_tree_link_by_ifn(1145):[lvl=0]Entered
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_compute_key(196):[lvl=0]type 2 slot
 0002 subunit 0001
*Sep 10 11:39:16.467:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_tree_link_by_ifn(1157):[lvl=0]computed key 0x2081FF01
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_state(3758):[lvl=0]endpt
aaln/S2/SU1/1
*Sep 10 11:39:16.467: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_default_get_state(3779):[lvl=0]endpt
 aaln/S2/SU1/1
*Sep 10 11:39:16.479: MGCP Packet sent to 192.168.1.200:7979--->
200 27 OK
I: D
v=0
c=IN IP4 192.168.1.79
m=audio 16870 RTP/AVP 0 8 99 101 102 2 15 103 4 104 105 106 107 18 100
a=rtpmap:99 G.729a/8000
a=rtpmap:101 G.726-16/8000
a=rtpmap:102 G.726-24/8000
a=rtpmap:103 G.723.1-H/8000
a=rtpmap:104 G.723.1-L/8000
a=rtpmap:105 G.729b/8000
a=rtpmap:106 G.723.1a-H/8000
a=rtpmap:107 G.723.1a-L/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 200-202
a=X-sqn:0
a=X-cap: 1 audio RTP/AVP 100
a=X-cpar: a=rtpmap:100 X-NSE/8000
a=X-cpar: a=fmtp:100 200-202
a=X-cap: 2 image udptl t38
<---
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp errors

To enable debug traces for Media Gateway Control Protocol (MGCP) errors, use the **debug mgcp errors**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp errors
no debug mgcp errors

Syntax Description

This command has no arguments or keywords.

Command Default

MGCP error debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1(3)T	Additional information was displayed for the gateways.
12.1(5)XM, 12.2(2)T	The output was modified to display parameters for the MGCP channel-associated signaling (CAS) PBX and ATM adaptation layer 2 (AAL2) permanent virtual circuit (PVC) features.
12.2(2)XA	The **endpoint** *endpoint-name*keyword and argument were added.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **endpoint** *endpoint-name*keyword and argument were removed and replaced by the **debug mgcp endpoint** command.

Usage Guidelines

This command enables error debugging globally for all MGCP endpoints. You can limit debugging to a specific endpoint by using the **debug mgcp endpoint** command.

Examples

The following is sample output from the **debug mgcp errors** command:

```
Router# debug mgcp errors
*Oct 16 12:09:05.538: MGC stat - 10.208.237.83, total=1029, succ=998, failed=0-
mgcp_parse_header()- Request Verb FOUND AUEP
- mgcp_parse_request_header()- MGCP_V10, start check for profile
- mgcp_parse_header: mgcp_parse_request_header returns status: 0
*Oct 16 12:09:05.538: MGCP Packet received from 10.208.237.83-
AUEP 9634549 S0/DS1-0/1@AS5300 MGCP 1.0
F: I
*Oct 16 12:09:05.542: -- mgcp_parse_packet() - call mgcp_parse_header
- mgcp_parse_header()- Request Verb FOUND AUEP
- mgcp_parse_request_header()- MGCP_V10, start check for profile
- mgcp_parse_header: mgcp_parse_request_header returns status: 0
- mgcp_parse_packet() - out mgcp_parse_header
```

```
- SUCCESS: mgcp_parse_packet()-MGCP Header parsing was OK
- mgcp_parse_parameter_lines(), code_str:: I, code_len:2, str:F: I
- mgcp_parse_parameter_lines(str:F: I) -num_toks: 28
- mgcp_parse_parameter_lines() check NULL str(I), in_ptr(F: I)
- mgcp_parse_parameter_lines() return Parse function in mgcp_parm_rules_array[14]
- mgcp_parse_req_info(I) is called
- mgcp_parse_req_info() - tmp_ptr:(I)
- SUCCESS: Request Info parameter line (F:) parsing OK
- mgcp_val_mandatory_parms()
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp events

To enable debug traces for Media Gateway Control Protocol (MGCP) events, use the **debug mgcp events**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp events [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp events

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP events debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1(3)T	Additional information was displayed for the gateways.
12.1(5)XM, 12.2(2)T	The output was modified to display parameters for the MGCP channel-associated signaling (CAS) PBX and ATM adaptation layer 2 (AAL2) permanent virtual circuit (PVC) features.
12.2(2)XA	The **endpoint** *endpoint-name*keyword and argument were added.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **endpoint** *endpoint-name* keyword and argument were removed and replaced by the **debug mgcp endpoint** command. The **tracelevel** keyword was added.

Usage Guidelines

This command enables events debugging globally for all MGCP endpoints. You can limit debugging to a specific endpoint by using the **debug mgcp endpoint** command.

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp events** command:

```
Router# debug mgcp events
Media Gateway Control Protocol events debugging for all endpoints is on, trace-level Verbose
Router#
*Sep 10 09:22:41.276: //-1/xxxxxxxxxxxx/MGCP/mgcpapp_stw_call_back(316):[lvl=0]timer type
1
*Sep 10 09:22:41.276: //-1/xxxxxxxxxxxx/MGCP/mgcpapp_process_timers(1431):[lvl=0]timer of
type 1 expired.
*Sep 10 09:22:41.276:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_remove_old_ack(712):[lvl=1]Removing ack:
(trans ID 15) : 250 15 OK
P: PS=0, OS=0, PR=0, OR=0, PL=0, JI=0, LA=0
*Sep 10 09:22:42.300: //-1/xxxxxxxxxxxx/MGCP/mgcp_count_active_mgc_msg_stat(240):[lvl=1]MGC
 stat - 192.168.1.200, total=18, succ=14, failed=2
*Sep 10 09:22:42.300: //-1/xxxxxxxxxxxx/MGCP/mgcpapp_process_mgcp_msg(3318):[lvl=0] : <NEW
 MGCP MSG From CA>
*Sep 10 09:22:42.300: //-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_endpt_offset(2590):[lvl=0]endpt
 NULL
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcpapp_setup_per_call_data(2487):[lvl=1]mgcpapp_setup_per_call_data:
 callp: 63E313E0, vdbptr: 65822AF8, state: CALL_IDLE
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_notified_entity(439):[lvl=0]Entered
*Sep 10 09:22:42.300: //-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_notified_entity(458):[lvl=1]ne
 callagenthost:7979, ne addr 192.168.1.200:7979
*Sep 10 09:22:42.300: //-1/xxxxxxxxxxxx/MGCP/xlate_mgcp_ev(921):[lvl=1]hdr_type 1
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcpapp_process_mgcp_event(2615):[lvl=1]Processing
 Incoming Message [CRCX 16]
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcpapp_invoke_mgcp_sm(2559):[lvl=1]Msg
In-Progress(Active) [INVVERB 0], await_ev=0, queued=0x00000000
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_process_deferred_queue(3362):[lvl=0]Entered
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP/mgcp_store_endpt_and_ntfy_entity_name(4464):[lvl=0]Entered
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_invoke_app_sm(535):[lvl=0]MGCP:calling FSM-
 callp(63E313E0)
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_idle_crcx(4875):[lvl=2]callp(0x63E313E0),
current state CALL_IDLE, event EV_CREATE_CONN
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP/mgcp_init_modem_relay_params(103):[lvl=0]modem-relay-enabled=0,
mr-gw-xid=0
*Sep 10 09:22:42.300:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_compute_debugsy_hdr(274):[lvl=0]Building
Debugsy header
*Sep 10 09:22:42.300:
//-1/C537F3F38008/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_compute_debugsy_hdr(383):[lvl=0]GUID[C537F3F38008]
 assigned to call_id[-1], endpt[aaln/S2/SU1/1], mgcp_call_id[n/a], conn_id[0]
*Sep 10 09:22:42.300:
//-1/C537F3F38008/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_idle_crcx(4961):[lvl=0]calls
mgcp_allocate_if()
*Sep 10 09:22:42.300:
//-1/C537F3F38008/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_idle_crcx(5006):[lvl=1]get capability
*Sep 10 09:22:42.300:
//-1/C537F3F38008/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_compute_debugsy_hdr(274):[lvl=0]Building
 Debugsy header
*Sep 10 09:22:42.304:
//-1/C537F3F38008/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_compute_debugsy_hdr(383):[lvl=0]GUID[C537F3F38008]
 assigned to call_id[-1], endpt[aaln/S2/SU1/1], mgcp_call_id[1], conn_id[0]
*Sep 10 09:22:42.304:
//-1/C537F3F38008/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_idle_crcx(5093):[lvl=0]Default aal2
```

```
 vc = 1 1-pvc,2-svc
*Sep 10 09:22:42.304: //-1/xxxxxxxxxxxx/MGCP/mgcp_init_vox_if_record(6781):[lvl=0]reusing
records. conn_type: 2, vox if_type: 1
*Sep 10 09:22:42.304: //-1/xxxxxxxxxxxx/MGCP/mgcp_compute_debugsy_hdr(274):[lvl=0]Building
 Debugsy header
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp gcfm

To enable generic call filter module (GCFM) debug traces for Media Gateway Control Protocol (MGCP), use the **debug mgcp gcfm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp gcfm [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp gcfm

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP GCFM debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(4)T	This command was introduced.

Usage Guidelines

This command enables GCFM debugging globally for all MGCP endpoints.

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp gcfm**command:

```
Router# debug mgcp gcfm
Media Gateway Control Protocol gcfm debugging for all endpoints is on, trace-level Verbose
Router#
*Sep 10 09:24:52.692:
//-1/12F030978009/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_gcfm_percall_register(315):[lvl=2]GCFM
 Inactive
*Sep 10 09:24:52.692:
//-1/12F030978009/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_gcfm_percall_register(315):[lvl=2]GCFM
 Inactive
Router#
```

Related Commands

Command	Description
debug call filter inout	Displays the debug trace inside the GCFM.

Command	Description
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp inout

To enable debug traces for all Media Gateway Control Protocol (MGCP) entry and exit endpoints, use the **debug mgcp inout**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp inout [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp inout

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

Debugging of MGCP entry and exit endpoints is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(4)T	This command was introduced.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp inout** command:

```
Router# debug mgcp inout
Media Gateway Control Protocol inout debugging for all endpoints is on, trace-level Verbose
Router#
*Sep 10 09:26:37.780: //-1/xxxxxxxxxxxx/MGCP/mgcp_count_active_mgc_msg_stat(240):[lvl=1]MGC
 stat - 192.168.1.200, total=22, succ=18, failed=2
*Sep 10 09:26:37.780: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(316):[lvl=0]call
mgcp_parse_header
*Sep 10 09:26:37.780: //-1/xxxxxxxxxxxx/MGCP/mgcp_string_parse(186):[lvl=0]return code=1.
*Sep 10 09:26:37.780: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4762):[lvl=0](in_ptr:
recvonly)
*Sep 10 09:26:37.780: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4894):[lvl=0]SUCCESS:
Connection Mode parsing is OK
*Sep 10 09:26:37.780: //-1/xxxxxxxxxxxx/MGCP/mgcp_string_parse(186):[lvl=0]return code=1.
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_call_id(840):[lvl=0]in_ptr: 1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_call_id(883):[lvl=1]SUCCESS: Call
ID string(1) parsing is OK
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_mandatory_parms(12428):[lvl=0]Entered
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_a(1339):[lvl=0]aaln/S2/SU1/1
```

```
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_slot(1632):[lvl=0]2/SU1/1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]2/SU1/1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_su(1773):[lvl=0]1/1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]1/1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_aaln_port(1807):[lvl=0]1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_name_parse_digit(1600):[lvl=0]1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_endpt_offset(2590):[lvl=0]endpt
 NULL
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_by_ifn(1326):[lvl=0]Entered
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_tree_link_by_ifn(1145):[lvl=0]Entered
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_compute_key(196):[lvl=0]type 2 slot
 0002 subunit 0001
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_get_state(3758):[lvl=0]endpt
aaln/S2/SU1/1
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xgcp_endpt_default_get_state(3779):[lvl=0]endpt
 aaln/S2/SU1/1
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_notified_entity(439):[lvl=0]Entered
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/mgcp_endpt_get_notified_entity(458):[lvl=1]ne
 callagenthost:7979, ne addr 192.168.1.200:7979
*Sep 10 09:26:37.784: //-1/xxxxxxxxxxxx/MGCP/xlate_mgcp_ev(921):[lvl=1]hdr_type 1
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcpapp_invoke_mgcp_sm(2559):[lvl=1]Msg
In-Progress(Active) [INVVERB 0], await_ev=0, queued=0x00000000
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_process_deferred_queue(3362):[lvl=0]Entered
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP/mgcp_store_endpt_and_ntfy_entity_name(4464):[lvl=0]Entered
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_idle_crcx(4875):[lvl=2]callp(0x63E313E0),
current state CALL_IDLE, event EV_CREATE_CONN
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP/mgcp_init_modem_relay_params(103):[lvl=0]modem-relay-enabled=0,
mr-gw-xid=0
*Sep 10 09:26:37.784:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcp_compute_debugsy_hdr(274):[lvl=0]Building
Debugsy header
*Sep 10 09:26:37.784:
//-1/5193F3E0800A/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_compute_debugsy_hdr(383):[lvl=0]GUID[5193F3E0800A]
 assigned to call_id[-1], endpt[aaln/S2/SU1/1], mgcp_call_id[n/a], conn_id[0]
*Sep 10 09:26:37.784:
//-1/5193F3E0800A/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_verify_supp_reqdet_ev(10645):[lvl=0]Entered
*Sep 10 09:26:37.784:
//-1/5193F3E0800A/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_verify_supp_signal_ev(10685):[lvl=0]Entered
*Sep 10 09:26:37.784:
//-1/5193F3E0800A/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_compute_debugsy_hdr(274):[lvl=0]Building
 Debugsy header
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.

Command	Description
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp media

To enable debug traces for Media Gateway Control Protocol (MGCP) tone and signal events, use the **debug mgcp media**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp media [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp media

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP media debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XA	This command was introduced.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **endpoint** *endpoint-name*keyword and argument were removed and replaced by the **debug mgcp endpoint** command. The **tracelevel** keyword was added.

Usage Guidelines

This command enables media debugging globally for all MGCP endpoints. You can limit debugging to a specific endpoint by using the **debug mgcp endpoint** command.

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp media** command:

```
Router# debug mgcp media

Media Gateway Control Protocol media events debugging for all endpoints is on, trace-level
 Verbose
Router#
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_verify_supp_reqdet_ev(10645):[lvl=0]Entered
*Sep 10 09:27:48.928:
```

```
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_verify_supp_signal_ev(10685):[lvl=0]Entered
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/process_request_ev(5800):[lvl=1]callp
63E313E0, voice_if 6663CA38
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/process_detect_ev(6007):[lvl=0]callp
63E313E0, voice_if 6663CA38
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/process_signal_ev(5500):[lvl=0]callp
63E313E0, voice_ifp 6663CA38
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_process_quarantine_mode(6096):[lvl=0]callp
 63E313E0, voice_if 6663CA38
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_process_quarantine_mode(6149):[lvl=0]Q
 mode not found, Reset default values
*Sep 10 09:27:48.928:
//-1/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_process_quarantine_mode(6168):[lvl=1]Q
 mode: process=0, loop=0
*Sep 10 09:27:48.936:
//19/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_process_pending_t38_port_switch(1649):[lvl=1]conn_recp->conn_id:
 0x0
*Sep 10 09:27:48.940:
//19/7BFBA9F9800B/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/process_deferred_request_events(5724):[lvl=0]Entered
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp nas

To enable network access server (NAS) (data) events for Media Gateway Control Protocol (MGCP), use the **debug mgcp nas**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp nas [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp nas

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP NAS event debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB	This command was introduced.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output for the **debug mgcp nas** command with the **debug mgcp packets** command also enabled:

```
Router# debug mgcp nas
Media Gateway Control Protocol nas pkg events debugging for all endpoints is on, trace-level
 Verbose
Router# debug mgcp packets
Media Gateway Control Protocol packets debugging for all endpoints is on
Router#
*Sep 10 11:51:41.863: MGCP Packet received from 192.168.1.200:7979--->
CRCX 34 aaln/S2/SU1/1 MGCP 1.0
X:57
M: nas/data
C:3
```

```
L:b:64, nas/bt:modem, nas/cdn:3000, nas/cgn:1000
C: 1
<---
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(779):[lvl=0]Full string:
nas/bt:modem
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(791):[lvl=1]string past slash:
 bt
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(792):[lvl=1]string past colon:
 modem
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(779):[lvl=0]Full string:
nas/cdn:3000
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(791):[lvl=1]string past slash:
 cdn
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(792):[lvl=1]string past colon:
 3000
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(779):[lvl=0]Full string:
nas/cgn:1000
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(791):[lvl=1]string past slash:
 cgn
*Sep 10 11:51:41.863: //-1/xxxxxxxxxxxx/MGCP/mgcp_chq_nas_pkg(792):[lvl=1]string past colon:
 1000
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp packets

To enable debug traces for Media Gateway Control Protocol (MGCP) packets, use the **debug mgcp packets**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp packets [**input-hex**]
no debug mgcp packets [**input-hex**]

Syntax Description

input-hex	(Optional) Displays MGCP incoming packets in hexadecimal format.

Command Default

MGCP packets debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1(3)T	Additional information was displayed for the gateways.
12.1(5)XM, 12.2(2)T	The output was modified to display parameters for the MGCP channel-associated signaling (CAS) PBX and ATM adaptation layer 2 (AAL2) permanent virtual circuit (PVC) features.
12.2(2)XA	The **endpoint** *endpoint-name*keyword and argument and the **input-hex** keyword were added.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **endpoint** *endpoint-name*keyword and argument were removed and replaced by the **debug mgcp endpoint** command.

Usage Guidelines

This command enables packet debugging globally for all MGCP endpoints. You can limit debugging to a specific endpoint by using the **debug mgcp endpoint** command.

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp packets** command:

```
Router# debug mgcp packets
Media Gateway Control Protocol packets debugging for all endpoints is on
Router#
*Sep 10 11:57:26.795: MGCP Packet received from 192.168.1.200:7979--->
CRCX 38 aaln/S2/SU1/1 MGCP 1.0
M: recvonly
```

```
C: 1
<---
*Sep 10 11:57:26.795:
//-1/xxxxxxxxxxxx/MGCP|aaln/S2/SU1/1|-1|-1/mgcpapp_invoke_mgcp_sm(2569):[lvl=0]CHECK DATA
CALL for aaln/S2/SU1/1
*Sep 10 11:57:26.807: MGCP Packet sent to 192.168.1.200:7979--->
200 38 OK
I: 10
v=0
c=IN IP4 192.168.1.79
m=audio 18876 RTP/AVP 0 8 99 101 102 2 15 103 4 104 105 106 107 18 100
a=rtpmap:99 G.729a/8000
a=rtpmap:101 G.726-16/8000
a=rtpmap:102 G.726-24/8000
a=rtpmap:103 G.723.1-H/8000
a=rtpmap:104 G.723.1-L/8000
a=rtpmap:105 G.729b/8000
a=rtpmap:106 G.723.1a-H/8000
a=rtpmap:107 G.723.1a-L/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 200-202
a=X-sqn:0
a=X-cap: 1 audio RTP/AVP 100
a=X-cpar: a=rtpmap:100 X-NSE/8000
a=X-cpar: a=fmtp:100 200-202
a=X-cap: 2 image udptl t38
<---
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp parser

To enable debug traces for the Media Gateway Control Protocol (MGCP) parser and builder, use the **debug mgcp parser** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp parser [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp parser

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP parser and builder debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.1(3)T	Additional information was displayed for the gateways.
12.1(5)XM, 12.2(2)T	The output was modified to display parameters for the MGCP channel-associated signaling (CAS) PBX and ATM adaptation layer 2 (AAL2) permanent virtual circuit (PVC) features.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp parser** command:

```
Router# debug mgcp parser

Media Gateway Control Protocol parser debugging for all endpoints is on, trace-level Verbose
Router#
```

```
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(316):[lvl=0]call
mgcp_parse_header
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(320):[lvl=0]out
mgcp_parse_header
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(360):[lvl=1]SUCCESS: - MGCP
 Header parsing was OK
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_string_parse(186):[lvl=0]return code=1.
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_parameter_lines(725):[lvl=1]return
 parse function in mgcp_parm_rules_array[6].
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4762):[lvl=0](in_ptr:
recvonly)
*Sep 10 11:58:51.283:
//-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4780):[lvl=0]tmp_ptr:(recvonly)
*Sep 10 11:58:51.283:
//-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4816):[lvl=0]tmp_ptr:(recvonly)
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4822):[lvl=0]match recvonly
 recvonly
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4830):[lvl=0]case
MODE_RECVONLY
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_conn_mode(4894):[lvl=0]SUCCESS:
Connection Mode parsing is OK
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_string_parse(186):[lvl=0]return code=1.
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_parameter_lines(725):[lvl=1]return
 parse function in mgcp_parm_rules_array[1].
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_call_id(840):[lvl=0]in_ptr: 1
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_call_id(883):[lvl=1]SUCCESS: Call
ID string(1) parsing is OK
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_mandatory_parms(12428):[lvl=0]Entered
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_comp_mp_parms(14923):[lvl=0]Entered
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_val_comp_mp_parms(14928):[lvl=1] -
lcon_opt_ptr could not be obtained
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_parse_packet(378):[lvl=2]SUCCESS: END of
 Parsing
*Sep 10 11:58:51.283:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_validate_version_with_call_agent_serv_type(8322):[lvl=1]req_msg_version:
 5, config_version: 5
*Sep 10 11:58:51.283: //-1/xxxxxxxxxxxx/MGCP/mgcp_validate_net_type(6601):[lvl=1]
lcnw_valid=0, lc_con_valid=0
*Sep 10 11:58:51.287: //-1/xxxxxxxxxxxx/MGCP/mgcp_validate_net_type(6710):[lvl=1]Network
type/conection type valid = 1. connection type = 1 [1->RTP, 2->AAL1_SDT, 4->AAL2]
*Sep 10 11:58:51.287: //-1/xxxxxxxxxxxx/MGCP/mgcp_get_qos(2665):[lvl=1]MGCP msg qos value=0
*Sep 10 11:58:51.287: //-1/xxxxxxxxxxxx/MGCP/mgcp_init_dyn_payload_types(2899):[lvl=1]used
 payload type map = 2F400003
*Sep 10 11:58:51.287: //-1/xxxxxxxxxxxx/MGCP/get_voip_peer_info(7155):[lvl=1]No SDP connection
 info
*Sep 10 11:58:51.287:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_select_codec_only(897):[lvl=1]num
supprt codec=14
*Sep 10 11:58:51.287:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_select_codec_only(1061):[lvl=0]glob
 codec=1 (syn=1)
*Sep 10 11:58:51.287:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_select_codec_only(1063):[lvl=0]supp
 list=
*Sep 10 11:58:51.287:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_select_codec_only(1067):[lvl=0] 1
*Sep 10 11:58:51.287:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_select_codec_only(1067):[lvl=0],2
*Sep 10 11:58:51.287:
//-1/95915C328011/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_select_codec_only(1067):[lvl=0],7
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp src

To enable debug traces for the System Resource Check (SRC) Call Admission Control (CAC) process for Media Gateway Control Protocol (MGCP), use the **debug mgcp src** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp src [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp src

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP SRC debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp src** command:

```
Router# debug mgcp src
Media Gateway Control Protocol System Resource Check CAC debugging for all endpoints is on,
 trace-level Verbose
Router#
*Sep 10 12:01:14.403:
//-1/EADF209C8013/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_set_call_counter_control(8163):[lvl=1]Outgoing
 call with 1 network leg, flag=TRUE
*Sep 10 12:03:01.051:
//35/EADF209C8013/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_reset_call_direction(8184):[lvl=1]Reseting
 incoming_call flag=FALSE in voice_if
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp state

To enable state traces for Media Gateway Control Protocol (MGCP), use the **debug mgcp state**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp state [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp state

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP state debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp state**command:

```
Router# debug mgcp state
Media Gateway Control Protocol state transition debugging for all endpoints is on, trace-level
 Verbose
Router#
*Sep 10 12:08:02.755:
//39/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E313E0)
 old state=CALL_IDLE new state=CALL_CONNECTING
*Sep 10 12:08:02.755:
//40/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E311D0)
 old state=CALL_IDLE new state=CALL_CONNECTING
*Sep 10 12:08:02.755:
//39/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E313E0)
 old state=CALL_CONNECTING new state=CALL_CONNECTING
*Sep 10 12:08:02.759:
//40/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E311D0)
 old state=CALL_CONNECTING new state=CALL_CONFERENCING
```

```
*Sep 10 12:08:02.759:
//39/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E313E0)
 old state=CALL_CONNECTING new state=CALL_CONFERENCING
*Sep 10 12:08:02.759:
//40/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E311D0)
 old state=CALL_CONFERENCING new state=CALL_CONFERENCING
*Sep 10 12:08:02.763:
//39/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E313E0)
 old state=CALL_CONFERENCING new state=CALL_ACTIVE
*Sep 10 12:08:02.763:
//40/DE454D0E8015/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_set_call_state(7562):[lvl=2]callp(0x63E311D0)
 old state=CALL_CONFERENCING new state=CALL_ACTIVE
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp tracelevel-default

To set the trace level globally for all Media Gateway Control Protocol (MGCP) debug traces, use the **debug mgcp tracelevel-default** command in privileged EXEC mode. To reset the trace level to the default value, use the **no** form of this command.

debug mgcp tracelevel-default {**critical** | **moderate** | **verbose**}
no debug mgcp tracelevel-default {**critical** | **moderate** | **verbose**}

Syntax Description

critical	Only high priority debug information is displayed.
moderate	Medium and high priority debug information is displayed.
verbose	All debug information is displayed. This is the default value.

Command Default

The default trace level for all MGCP debug commands is verbose.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(4)T	This command was introduced.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

This command sets the trace level globally for all MGCP debug commands and endpoints. You can set the trace level independently for a specific endpoint by using the **debug mgcp endpoint** command. The endpoint-specific trace level takes precedence over the global trace-level set with this command.

Note This command applies only to MGCP debug commands that are issued after the default trace level is set. For example, if you enable several debug commands and then change the default trace level, the new trace level does not apply to any previously enabled MGCP debug commands.

Examples

The following example sets the default trace level to critical for all MGCP debug traces:

```
Router# debug mgcp tracelevel-default critical
Router# debug mgcp events

Media Gateway Control Protocol events debugging for all endpoints is on, trace-level Critical
Router# debug mgcp state

Media Gateway Control Protocol state transition debugging for all endpoints is on, trace-level
 Critical
```

Notice that if the default trace level is then changed, as in the following example, the new trace level applies only to any MGCP debug commands that are issued after the default trace level is changed.

```
Router# debug mgcp tracelevel-default verbose

Router# debug mgcp voipcac

Media Gateway Control Protocol VoIPCAC debugging for all endpoints is on, trace-level Verbose
Router# show debug
MGCP:
  Media Gateway Control Protocol events debugging is on, trace level Critical
  Media Gateway Control Protocol VoIPCAC debugging is on, trace level Verbose
  Media Gateway Control Protocol state transition debugging is on, trace level Critical
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp endpoint	Enables debug traces for a specific MGCP endpoint.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mgcp voipcac

To enable debug traces for the Voice over IP (VoIP) Call Admission Control (CAC) process at the Media Gateway Control Protocol (MGCP) application layer, use the **debug mgcp voipcac** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mgcp voipcac [**tracelevel** {**critical** | **moderate** | **verbose**}]
no debug mgcp voipcac

Syntax Description

tracelevel	(Optional) Sets the priority level for this debug trace. • **critical** --Displays only high priority debug information. • **moderate** --Displays medium and high priority debug information. • **verbose** --Displays all debug information. This is the default level.

Command Default

MGCP VoIP CAC debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.2(11)T	The command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850.
12.2(13)T	Support for this command was implemented in Cisco 7200 series images.
12.4(4)T	The **tracelevel** keyword was added.

Usage Guidelines

Trace levels allow you to control the amount of debug information that is displayed in the output from MGCP debug commands. Reducing the amount of output displayed on the console port makes it easier to locate the correct debug information and limits the impact to network performance.

Examples

The following is sample output from the **debug mgcp voipcac**command:

```
Router# debug mgcp voipcac
Media Gateway Control Protocol VoIPCAC debugging for all endpoints is on, trace-level Verbose
Router#
*Sep 10 12:04:47.747:
//-1/6A09713E8014/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_idle_crcx(5251):[lvl=0]Check for
HP and QOS combination
*Sep 10 12:04:47.751:
//-1/6A09713E8014/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/mgcp_idle_crcx(5451):[lvl=0]CAC success
*Sep 10 12:04:47.751:
//-1/6A09713E8014/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/set_up_voip_call_leg(3918):[lvl=0]get
```

```
voice interface
*Sep 10 12:04:47.751:
//-1/6A09713E8014/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/set_up_voip_call_leg(4441):[lvl=0]Initialize
 VoIP CAC record stored in
                    VoIP interface struct
*Sep 10 12:04:47.751:
//38/6A09713E8014/MGCP|aaln/S2/SU1/1|-1|-1/<VOIP>/mgcp_connect_peer_vox_call_leg(1546):[lvl=0]set_up_voip_call_leg
 returns OK
*Sep 10 12:04:47.759:
//37/6A09713E8014/MGCP|aaln/S2/SU1/1|-1|-1/<VOICE>/process_signal_request_list(5608):[lvl=0]Entered
```

Related Commands

Command	Description
debug mgcp all	Enables all debug traces for MGCP.
debug mgcp tracelevel-default	Sets the trace level globally for all MGCP debug traces.
mgcp	Starts the MGCP daemon.
mgcp debug-header	Enables the display of MGCP module-dependent information in the debug header.
show mgcp	Displays MGCP configuration information.
voice call debug	Specifies the format of the debug header.

debug mlrib common

To enable logging of common Multilayer Routing Information Base (MLRIB) debug messages, use the **debug mlrib common** command in privileged EXEC mode. To disable the debugging, use the **no** form of this command.

debug mlrib common {**event** {**all** | **client_api** | **db** | **ha** | **misc** | **notify** | **registration** | **show** | **srw** | **utils**} | **error**}
no debug mlrib common {**event** {**all** | **client_api** | **db** | **ha** | **misc** | **notify** | **registration** | **show** | **srw** | **utils**} | **error**}

Syntax Description

event	Enables logging of event debug messages.
all	Enables logging of all common debug events.
client_api	Enables client API-related debugging.
db	Enables MLRIB database debugging.
ha	Enables MLRIB high availability (HA) debugging.
misc	Enables miscellaneous events debugging.
notify	Enables MLRIB notify debugging.
registration	Enables MLRIB registration-related debugging.
show	Enables debugging of MLRIB **show** commands.
srw	Enables MLRIB Single Reader Writer (SRW) debugging.
utils	Enables MLRIB utilities debugging.
error	Enables error debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following example shows how to enable debugging of common MLRIB events:

```
Router# debug mlrib common event all

MLRIB COMMON event all debugging is on
*Oct 28 07:26:54.614: MLRIB_COMMON_REGISTRATION: client state set: ISISL2 OTV Overlay2
moving to REGISTERED state
*Oct 28 07:26:54.614: MLRIB_COMMON_REGISTRATION: client state set: ISISL2 OTV Overlay2
moving to REGISTERED state
```

Related Commands

Command	Description
show OTV	Displays information about OTV.

debug mlrib layer2

To enable logging of Layer 2-specific Multilayer Routing Information Base (MLRIB) debug messages, use the **debug mlrib layer2** command in privileged EXEC mode. To disable the debugging, use the **no** form of this command.

debug mlrib layer2 {**event** {**add** | **all** | **delete** | **flush** | **notify** | **redistribute** | **registration**} | **error**}
no debug mlrib layer2 {**event** {**add** | **all** | **delete** | **flush** | **notify** | **redistribute** | **registration**} | **error**}

Syntax Description

event	Enables logging of Layer 2 event debug messages.
add	Enables logging of Layer 2 add MLRIB debug events.
all	Enables logging of all Layer 2 MLRIB debug events.
delete	Enables logging of Layer 2 delete MLRIB debug events.
flush	Enables logging of Layer 2 flush MLRIB debug events.
notify	Enables logging of Layer 2 notify MLRIB debug events.
redistribute	Enables logging of Layer 2 redistribution MLRIB debug events.
registration	Enables logging of Layer 2 registration MLRIB debug events.
error	Enables logging of Layer 2 error debug messages.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following example shows how to enable debugging of Layer 2 MLRIB events:

```
Router# debug mlrib layer2 event all

MLRIB L2 event all debugging is on
*Oct 28 07:25:23.257: MLRIB_L2_FLUSH: u flush req msg: flush notifications sent for pp=0x8,
 topo=10
*Oct 28 07:25:23.257: MLRIB_L2_FLUSH: u flush req msg: complete for pp=0x8, topo=12, client
 ISISL2 OTV Overlay1
*Oct 28 07:25:23.257: MLRIB_L2_FLUSH: u flush req msg: flush notifications sent for pp=0x8,
 topo=12
*Oct 28 07:25:23.257: MLRIB_L2_REDISTRIBUTE: hndl ucast redist refresh msg: Rcvd msg length
 20, redist id = 0x0 walk id 1102745848client = ISISL2 OTV Overlay1
*Oct 28 07:25:23.257: MLRIB_L2_REDISTRIBUTE: hndl ucast redist refresh msg: found filter
for redist id = 0x0
*Oct 28 07:25:23.257: MLRIB_L2_REDISTRIBUTE: redist walk setup: for vpn 0x1 and client
ISISL2 OTV Overlay1
```

Related Commands

Command	Description
show OTV	Displays information about OTV.

debug mls rp

To display various Internetwork Packet Exchange (IPX) Multilayer Switching (MLS) debugging elements, use the **debug mls rp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mls rp {**error** | **events** | **ipx** | **locator** | **packets** | **all**}
no debug mls rp {**error** | **events** | **ipx** | **locator** | **packets** | **all**}

Syntax Description

error	Displays MLS error messages.
events	Displays a run-time sequence of events for the Multilayer Switching Protocol (MLSP).
ipx	Displays IPX-related events for MLS, including route purging and changes to access lists and flow masks.
locator	Identifies which switch is switching a particular flow of MLS explorer packets.
packets	Displays packet contents (in verbose and hexadecimal formats) for MLSP messages.
all	Displays all MLS debugging events.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following shows sample output from the **debug mls rp ipx** command:

```
Router# debug mls rp ipx
IPX MLS debugging is on
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int vlan 22
Router(config-if)# no ipx access-group out
05:44:37:FCP:flowmask changed to destination
```

Related Commands

Command	Description
debug dss ipx event	Displays debugging messages for route change events that affect IPX MLS.

debug mls rp ip multicast

To display information about Multilayer Switching Protocol (MLSP), use the **debug mls rp ip multicast**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mls rp ip multicast {**all** | **error** | **events** | **packets**}
no debub mls rp ip multicast {**all** | **error** | **events** | **packets**}

Syntax Description

all	Displays all multicast MLSP debugging information, including errors, events, and packets.
error	Displays error messages related to multicast MLSP.
events	Displays the run-time sequence of events for multicast MLSP.
packets	Displays the contents of MLSP packets.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug mls rp ip multicast**command using the **error** keyword:

```
Router# debug mls rp ip multicast error
mlsm error debugging is on
chtang-7200#
06:06:45: MLSMERR: scb is INACTIVE, free INSTALL_FE
06:06:46: MLSM: --> mlsm_proc_sc_ins_req(10.0.0.1, 224.2.2.3, 10)
```

The following example shows output from the **debug mls rp ip multicast**command using the **event** keyword:

```
Router# debug mls rp ip multicast event
mlsm events debugging is on
Router#
3d23h: MSCP: incoming shortcut flow statistic  from Fa2/0.11
3d23h: MLSM: Flow_stat: (192.1.10.6, 239.255.158.197), byte :537792
packet:8403
3d23h: MLSM: byte delta:7680 packet delta:120, time delta: 10
3d23h: MSCP: incoming shortcut flow statistic  from Fa2/0.11
3d23h: MLSM: Flow_stat: (192.1.10.6, 239.255.158.197), byte :545472
packet:8523
3d23h: MLSM: byte delta:7680 packet delta:120, time delta: 10
3d23h: MSCP: Router transmits keepalive_msg on Fa2/0.11
```

```
3d23h: MSCP: incoming shortcut keepalive ACK  from Fa2/0.11
3d23h: MLSM: Include-list: (192.1.2.1  ->  0.0.0.0)
3d23h: MSCP: incoming shortcut flow statistic  from Fa2/0.11
3d23h: MLSM: Flow_stat: (192.1.10.6, 239.255.158.197), byte :553152
packet:8643
```

The following example shows output from the **debug mls rp ip multicast**command using the **packet** keyword:

```
Router# debug mls rp ip multicast packet
mlsm packets debugging is on
Router#
Router#
Router#
Router#
**23h: MSCP(I): 01 00 0c cc cc cc 00 e0 1e 7c fe 5f 00 30 aa aa
...LLL.`.|~_.0
..23h: MSCP(I): 03 00 00 0c 01 07 01 05 00 28 01 02 0a c7 00 10
.........(...G
 ..23h: MSCP(I): a6 0b b4 ff 00 00 c0 01 0a 06 ef ff 9e c5 00 00
&.4...@...o..E
3d23h: MSCP(I): 00 00 00 09 42 c0 00 00 00 00 00 00 25 0b
....B@......%.
3d23h:
**23h: MSCP(O): 01 00 0c 00 00 00 aa 00 04 00 01 04 00 00 aa aa
......*.......
LL23h: MSCP(O): 03 00 00 0c 00 16 00 00 00 00 01 00 0c cc cc cc
.............L
..23h: MSCP(O): aa 00 04 00 01 04 00 24 aa aa 03 00 00 0c 01 07
*......$**....
..23h: MSCP(O): 01 06 00 1c c0 01 02 01 aa 00 04 00 01 04 00 00
....@...*.....
3d23h: MSCP(O): 00 0b 00 00 00 00 00 00 01 01 0a 62 ...........b
3d23h:
**23h: MSCP(I): 01 00 0c cc cc cc 00 e0 1e 7c fe 5f 00 24 aa aa
...LLL.`.|~_.$
..23h: MSCP(I): 03 00 00 0c 01 07 01 86 00 1c 01 02 0a c7 00 10
.............G
..23h: MSCP(I): a6 0b b4 ff 00 00 00 0b 00 00 c0 01 02 01 00 00
..4.......@...
3d23h: MSCP(I): 00 00
3d23h:
```

Related Commands

Command	Description
debug mdss	Displays information about MDSS.

debug mmoip aaa

Note Effective with release 12.3(8)T, the **debug mmoip aaa**command is replaced by the **debug fax mmoip aaa**command. See the **debug fax mmoip aaa**command for more information.

To display output that relates to authentication, authorization, and accounting (AAA) services with store-and-forward fax, use the **debug mmoip aaa** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mmoip aaa
no debug mmoip aaa

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.2(4)T	This command was implemented on the Cisco 1750 access router.
12.3(8)T	This command was replaced by the **debug fax mmoip aaa** command.

Examples

The following output shows how the **debug mmoip aaa** command provides information about AAA for the on-ramp or off-ramp gateways:

```
Router# debug mmoip aaa
5d10h:fax_aaa_begin_authentication:User-Name = mmoip-b.cisco.com
5d10h:fax_aaa_begin_authentication:fax_account_id_origin = GATEWAY_ID
5d10h:fax_aaa_end_authentication_callback:Authentication successful
```

The following output shows how the **debug mmoip aaa** command provides information about AAA for the off-ramp gateway:

```
Router# debug mmoip aaa
5d10h:fax_aaa_start_accounting:User-Name = mmoip-b.cisco.com
5d10h:fax_aaa_start_accounting:Calling-Station-Id = gmercuri@mail-server.cisco.com
5d10h:fax_aaa_start_accounting:Called-Station-Id = fax=571-0839@mmoip-b.cisco.com
5d10h:fax_aaa_start_accounting:fax_account_id_origin = GATEWAY_ID
mmoip-b#ax_aaa_start_accounting:fax_msg_id = <37117AF3.3D98300E@mail-server.cisco.com>
5d10h:fax_aaa_start_accounting:fax_pages = 2
5d10h:fax_aaa_start_accounting:fax_coverpage_flag = TRUE
5d10h:fax_aaa_start_accounting:fax_connect_speed = 14400bps
5d10h:fax_aaa_start_accounting:fax_recipient_count = 1
5d10h:fax_aaa_start_accounting:fax_auth_status = USER SUCCESS
5d10h:fax_aaa_start_accounting:gateway_id = mmoip-b.cisco.com
5d10h:fax_aaa_start_accounting:call_type = Fax Send
```

```
5d10h:fax_aaa_start_accounting:port_used =  slot:0 vfc port:0
5d10h:fax_aaa_do_offramp_accounting tty(6), Stopping accounting
5d10h:fax_aaa_stop_accounting:ftdb->cact->generic.callActiveTransmitBytes = 18038
5d10h:fax_aaa_stop_accounting:ftdb->cact->generic.callActiveTransmitPackets = 14
```

The following output shows how the **debug mmoip aaa** command provides information about AAA for the on-ramp gateway:

```
Router# debug mmoip aaa
5d10h:fax_aaa_start_accounting:User-Name = mmoip-b.cisco.com
5d10h:fax_aaa_start_accounting:Calling-Station-Id = FAX=408@mail-from-hostname.com
5d10h:fax_aaa_start_accounting:Called-Station-Id = FAX=5710839@mail-server.cisco.com
5d10h:fax_aaa_start_accounting:fax_account_id_origin = GATEWAY_ID
5d10h:fax_aaa_start_accounting:fax_msg_id = 00391997233216263@mmoip-b.cisco.com
5d10h:fax_aaa_start_accounting:fax_pages = 2
5d10h:fax_aaa_start_accounting:fax_connect_speed = 14400bps
5d10h:fax_aaa_start_accounting:fax_auth_status = USER SUCCESS
5d10h:fax_aaa_start_accounting:email_server_address = 1.14.116.1
5d10h:fax_aaa_start_accounting:email_server_ack_flag = TRUE
5d10h:fax_aaa_start_accounting:gateway_id = mmoip-b.cisco.com
5d10h:fax_aaa_start_accounting:call_type = Fax Receive
5d10h:fax_aaa_start_accounting:port_used = Cisco Powered Fax System slot:1 port:4
5d10h:fax_aaa_do_onramp_accounting tty(5), Stopping accounting
5d10h:fax_aaa_stop_accounting:endb->cact->generic.callActiveTransmitBytes = 26687
5d10h:fax_aaa_stop_accounting:ftdb->cact->generic.callActiveReceiveBytes = 18558
5d10h:fax_aaa_stop_accounting:ftdb->cact->generic.callActiveReceivePackets = 14
```

debug mmoip send email

To test connectivity between the T.37 on-ramp gateway and the e-mail server by sending a test e-mail to a specified e-mail address, use the **debug mmoip send email** command in privileged EXEC mode.

debug mmoip send email *string*

Syntax Description

string	E-mail address of the sender; for example, mailuser@mail-server.com. There is no default.

Command Default

This command is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.2(4)T	This command was introduced on the Cisco 1750 access router.
12.2(8)T	This command was implemented on the Cisco 1751 access routers, Cisco 3725 access routers, and Cisco 3745 access routers.
12.2(13)T	This feature was implemented on the Cisco 7200 series routers.

Examples

The **debug mmoip send email** command is used to test connectivity between the on-ramp gateway and the e-mail server. Basically, this **debug** command sends an e-mail message to the recipient specified in the e-mail address string. There is no specific output associated with the **debug mmoip send email** command; to see how the on-ramp gateway and e-mail server interact when processing the test e-mail message, enable the **debug fmail client** command.

The following example tests connectivity between the on-ramp gateway and the e-mail server by sending a test e-mail message to mailuser@mail-server.com:

```
Router#
debug fmail client

Router#
debug mmoip send email mailuser@mail-server.com
01:22:59:faxmail_client_send_test:Sending the test message to
ilya@mail-server.com from testing@mmoip-a.cisco.com...
01:22:59:faxmail_client_send_test:Opening client engine.
01:22:59:faxmail_client_send_test:Sending 59 bytes ...
01:22:59:faxmail_client_send_test:Done sending test email.
```

Related Commands

Command	Description
debug fmail client	Displays e-mail parameters (such as Mail from and Envelope to and Envelope from) and the progress of the SMTP client.

debug mmoip send fax

To send a T.37 off-ramp test fax, use the **debug mmoip send fax** command in privileged EXEC mode.

debug mmoip send fax *string*

Syntax Description

string	E.164 telephone number to be used for sending the test fax. There is no default.

Command Default

This command is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.2(4)T	This command was implemented on the Cisco 1750 access router.
12.2(8)T	This command was implemented on the Cisco 1751 access routers, Cisco 3725 access routers, and Cisco 3745 access routers.
12.2(13)T	This feature was introduced on the Cisco 7200 series routers.

Examples

The **debug mmoip send fax** command is used to test connectivity between the off-ramp gateway and a recipient fax device. Basically, this **debug** command sends a test fax transmission to the recipient specified in the telephone number string. There is no specific output associated with the **debug mmoip send fax** command.

The following example sends a test fax message to the telephone number 5550839:

```
Router# debug mmoip send fax 5550839
```

The following output shows that the off-ramp gateway is placing a fax call:

```
01:28:18:ftsp_offramp_match_digits:phone number to translate:5550839
01:28:18: destPat(5......), matched(1), prefix() peer_tag(1)
01:28:18:ftsp_offramp_match_digits:target:710839
01:28:18:fap_offcm:tty(4), Got dial message00:00:00.000:AT&F\Q0S7=255
```

Class 2 modem tracing begins, including modem initialization.

```
00:00:00.008:AA
00:00:00.068:TT
00:00:00.128:&F\Q0S7=255
00:00:00.128:
OK

00:00:00.128:E0V1
00:00:00.140:ATE0
OK

00:00:00.140:AT+FCLASS=2
```

```
00:00:00.148:
OK

00:00:00.148:+FDCC=..;+FBOR=
00:00:00.168:AT+FLID
00:00:00.180:
OK

00:00:00.180:ATDTW710839
The following output shows that the fax transmission is complete; in this particular example,
 there was a transmission error, and the modem timed out.
01:28:25:ftsp_setup_for_oc:tty4, callid=0xA
01:28:25:ftsp_setup_for_oc ctl=0, cas grp=-1, snmp_ix=30
01:28:25:ftsp_off_ramp_active_call_init tty4 callid=0xA, snmp_ix=30
01:29:18:fap_offpmt:tty(4), TxPhaseA:modem timeout
01:29:18:%FTSP-6-FAX_DISCONNECT:Transmission er
```

debug mmoip transfer

To send output of the Tag Image File Format (TIFF) writer to a TFTP server, use the **debug mmoip transfer** command in privileged EXEC mode.

debug mmoip transfer *prefix-filename tftp-server-name*

Syntax Description

prefix-filename	Name of the TIFF file. The format for the TIFF filename is "telephone-number.TIFF."
tftp-server-name	TFTP server to which the output from the TIFF writer is sent.

Command Default

Sending output of the TIFF writer to a TFTP server is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.2(4)T	This command was implemented on the Cisco 1750 access router.

Examples

The **debug mmoip transfer command**sends the content of the fax data received to the TFTP server named by the *tftp-server-name* variable into the file identified by the *prefix-filename* variable. Each page of the fax transmission is a separate file, designated by the letter "p", followed by the page number.

For example, the following command transfers the received fax content to a TFTP server named "keyer". The first page of the transmission goes to the file named "/tftpboot/test/testp1.tiff", the second page goes to the file named "/tftpboot/test/testp2.tiff" and so on.

```
Router# debug mmoip transfer /tftpboot/test/test keyer
```

The named files must exist on the TFTP server and be writable in order for the debug operation to be successful.

debug modem

To observe modem line activity on an access server, use the **debug modem** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem
no debug modem

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debug modem** command. The output shows when the modem line changes state.

```
Router# debug modem
15:25:51: TTY4: DSR came up
15:25:51: tty4: Modem: IDLE->READY
15:25:51: TTY4: Autoselect started
15:27:51: TTY4: Autoselect failed
15:27:51: TTY4: Line reset
15:27:51: TTY4: Modem: READY->HANGUP
15:27:52: TTY4: dropping DTR, hanging up
15:27:52: tty4: Modem: HANGUP->IDLE
15:27:57: TTY4: restoring DTR
15:27:58: TTY4: DSR came up
```

debug modem csm

To debug the Call Switching Module (CSM), used to connect calls on the modem, use the **debug modem csm**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem csm [{*slot/port* | **group** *group-number*}]
no debug modem csm [{*slot/port* | **group** *group-number*}]

Syntax Description

slot/port	(Optional) The slot and modem port number.
group *group-number*	(Optional) The modem group.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug modem csm** command to troubleshoot call switching problems. With this command, you can trace the complete sequence of switching incoming and outgoing calls.

Examples

The following is sample output from the **debug modem csm** command. In this example, a call enters the modem (incoming) on slot 1, port 0:

```
Router(config)# service timestamps debug uptime
Router(config)# end
Router# debug modem csm
00:04:09: ccpri_ratetoteup bear rate is 10
00:04:09: CSM_MODEM_ALLOCATE: slot 1 and port 0 is allocated.
00:04:09: MODEM_REPORT(0001): DEV_INCALL at slot 1 and port 0
00:04:09: CSM_PROC_IDLE: CSM_EVENT_ISDN_CALL at slot 1, port 0
00:04:11: CSM_RING_INDICATION_PROC: RI is on
00:04:13: CSM_RING_INDICATION_PROC: RI is off
00:04:15: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
00:04:15: MODEM_REPORT(0001): DEV_CONNECTED at slot 1 and port 0
00:04:15: CSM_PROC_IC2_WAIT_FOR_CARRIER: CSM_EVENT_ISDN_CONNECTED at slot 1, port 0
```

The following is sample output from the **debug modem csm**command when call is dialed from the modem into the network (outgoing) from slot 1, port 2:

```
Router# debug modem csm
atdt16665202
00:11:21: CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 2
00:11:21: T1_MAIL_FROM_NEAT: DC_READY_RSP: mid = 1, slot = 0, unit = 0
00:11:21: CSM_PROC_OC1_REQUEST_DIGIT: CSM_EVENT_DIGIT_COLLECT_READY at slot 1, port 2
00:11:24: T1_MAIL_FROM_NEAT: DC_FIRST_DIGIT_RSP: mid = 1, slot = 0, unit = 0
00:11:24: CSM_PROC_OC2_COLLECT_1ST_DIGIT: CSM_EVENT_GET_1ST_DIGIT at slot 1, port 2
00:11:27: T1_MAIL_FROM_NEAT: DC_ALL_DIGIT_RSP: mid = 1, slot = 0, unit = 0
00:11:27: CSM_PROC_OC3_COLLECT_ALL_DIGIT: CSM_EVENT_GET_ALL_DIGITS (16665202) at slot 1,
port 2
00:11:27: ccpri_ratetoteup bear rate is 10
00:11:27: MODEM_REPORT(A000): DEV_CALL_PROC at slot 1 and port 2
00:11:27: CSM_PROC_OC4_DIALING: CSM_EVENT_ISDN_BCHAN_ASSIGNED at slot 1, port 2
00:11:31: MODEM_REPORT(A000): DEV_CONNECTED at slot 1 and port 2
00:11:31: CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_ISDN_CONNECTED at slot 1, port 2
CONNECT 19200/REL - MNP
```

The following is sample output from the **debug modem csm** command for an incoming call:

```
Router# debug modem csm
Router#1.19.36.7 2001
Trying 1.19.36.7, 2001 ... Open
atdt111222333444555666
*Apr  7 12:39:42.475: Mica Modem(1/0): Rcvd Dial String(111222333444555666)
*Apr  7 12:39:42.475: CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
*Apr  7 12:39:42.479: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_CHANNEL_LOCK at slot 1 and
 port 0
*Apr  7 12:39:42.479: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_BCHAN_ASSIGNED at slot 1, port
0
*Apr  7 12:39:42.479: Mica Modem(1/0): Configure(0x1)
*Apr  7 12:39:42.479: Mica Modem(1/0): Configure(0x5)
*Apr  7 12:39:42.479: Mica Modem(1/0): Call Setup
*Apr  7 12:39:42.479: neat msg at slot 0: (1/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.491: neat msg at slot 0: (0/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.531: VDEV_ALLOCATE: slot 1 and port 3 is allocated.
*Apr  7 12:39:42.531: CSM_RX_CAS_EVENT_FROM_NEAT:(0004):  EVENT_CALL_DIAL_IN at slot 1 and
 port 3
*Apr  7 12:39:42.531: CSM_PROC_IDLE: CSM_EVENT_DSX0_CALL at slot 1, port 3
*Apr  7 12:39:42.531: Mica Modem(1/3): Configure(0x0)
*Apr  7 12:39:42.531: Mica Modem(1/3): Configure(0x5)
*Apr  7 12:39:42.531: Mica Modem(1/3): Call Setup
*Apr  7 12:39:42.595: Mica Modem(1/0): State Transition to Call Setup
*Apr  7 12:39:42.655: Mica Modem(1/3): State Transition to Call Setup
*Apr  7 12:39:42.655: Mica Modem(1/3): Went offhook
*Apr  7 12:39:42.655: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 3
*Apr  7 12:39:42.671: neat msg at slot 0: (0/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.691: neat msg at slot 0: (1/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:42.731: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_START_TX_TONE at slot 1 and
 port 0
*Apr  7 12:39:42.731: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0
*Apr  7 12:39:42.731: Mica Modem(1/0): Generate digits:called_party_num= len=1
*Apr  7 12:39:42.835: Mica Modem(1/3): Rcvd Digit detected(#)
*Apr  7 12:39:42.835: CSM_PROC_IC2_COLLECT_ADDR_INFO: CSM_EVENT_KP_DIGIT_COLLECTED (DNIS=,
 ANI=) at slot 1, port 3
*Apr  7 12:39:42.855: neat msg at slot 0: (0/0): Tx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:42.871: neat msg at slot 0: (1/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:42.899: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:42.911: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_END_TX_TONE at slot 1 and
port 0
*Apr  7 12:39:42.911: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_END_TX_TONE at slot 1, port 0
*Apr  7 12:39:42.911: Mica Modem(1/0): Generate digits:called_party_num=A len=1
*Apr  7 12:39:43.019: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:43.019: CSM_PROC_OC4_DIALING: CSM_EVENT_TONE_GENERATED at slot 1, port 0
*Apr  7 12:39:43.019: Mica Modem(1/3): Rcvd Digit detected(A)
*Apr  7 12:39:43.335: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_START_TX_TONE at slot 1 and
 port 0
*Apr  7 12:39:43.335: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0
*Apr  7 12:39:43.335: Mica Modem(1/0): Generate digits:called_party_num=111222333444555666
 len=19
*Apr  7 12:39:43.439: Mica Modem(1/3): Rcvd Digit detected(1)
*Apr  7 12:39:43.559: Mica Modem(1/3): Rcvd Digit detected(1)
*Apr  7 12:39:43.619: Mica Modem(1/3): Rcvd Digit detected(1)
*Apr  7 12:39:43.743: Mica Modem(1/3): Rcvd Digit detected(2)
*Apr  7 12:39:43.859: Mica Modem(1/3): Rcvd Digit detected(2)
*Apr  7 12:39:43.919: Mica Modem(1/3): Rcvd Digit detected(2)
*Apr  7 12:39:44.043: Mica Modem(1/3): Rcvd Digit detected(3)
*Apr  7 12:39:44.163: Mica Modem(1/3): Rcvd Digit detected(3)
*Apr  7 12:39:44.223: Mica Modem(1/3): Rcvd Digit detected(3)
*Apr  7 12:39:44.339: Mica Modem(1/3): Rcvd Digit detected(4)
*Apr  7 12:39:44.459: Mica Modem(1/3): Rcvd Digit detected(4)
*Apr  7 12:39:44.523: Mica Modem(1/3): Rcvd Digit detected(4)
*Apr  7 12:39:44.639: Mica Modem(1/3): Rcvd Digit detected(5)
```

```
*Apr  7 12:39:44.763: Mica Modem(1/3): Rcvd Digit detected(5)
*Apr  7 12:39:44.883: Mica Modem(1/3): Rcvd Digit detected(5)
*Apr  7 12:39:44.943: Mica Modem(1/3): Rcvd Digit detected(6)
*Apr  7 12:39:45.063: Mica Modem(1/3): Rcvd Digit detected(6)
*Apr  7 12:39:45.183: Mica Modem(1/3): Rcvd Digit detected(6)
*Apr  7 12:39:45.243: Mica Modem(1/3): Rcvd Digit detected(B)
*Apr  7 12:39:45.243: CSM_PROC_IC2_COLLECT_ADDR_INFO: CSM_EVENT_DNIS_COLLECTED
(DNIS=111222333444555666, ANI=) at slot 1, port 3
*Apr  7 12:39:45.363: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:45.891: neat msg at slot 0: (0/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:45.907: neat msg at slot 0: (1/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:46.115: neat msg at slot 0: (0/0): Tx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:46.131: neat msg at slot 0: (1/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:39:46.175: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_START_TX_TONE at slot 1 and
 port 0
*Apr  7 12:39:46.175: CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0
*Apr  7 12:39:46.175: Mica Modem(1/0): Generate digits:called_party_num= len=3
*Apr  7 12:39:46.267: Mica Modem(1/3): Rcvd Digit detected(#)
*Apr  7 12:39:46.387: Mica Modem(1/3): Rcvd Digit detected(A)
*Apr  7 12:39:46.447: Mica Modem(1/3): Rcvd Digit detected(B)
*Apr  7 12:39:46.447: CSM_PROC_IC2_COLLECT_ADDR_INFO: CSM_EVENT_ADDR_INFO_COLLECTED
(DNIS=111222333444555666, ANI=) at slot 1, port 3
*Apr  7 12:39:46.507: Mica Modem(1/0): Rcvd Digits Generated
*Apr  7 12:39:46.507: CSM_PROC_OC4_DIALING: CSM_EVENT_ADDR_INFO_COLLECTED at slot 1, port
0
*Apr  7 12:39:47.127: CSM_RX_CAS_EVENT_FROM_NEAT:(0004):  EVENT_CHANNEL_CONNECTED at slot
1 and port 3
*Apr  7 12:39:47.127: CSM_PROC_IC4_WAIT_FOR_CARRIER: CSM_EVENT_DSX0_CONNECTED at slot 1,
port 3
*Apr  7 12:39:47.127: Mica Modem(1/3): Link Initiate
*Apr  7 12:39:47.131: neat msg at slot 0: (0/0): Tx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:47.147: neat msg at slot 0: (1/0): Rx LOOP_CLOSURE (ABCD=1101)
*Apr  7 12:39:47.191: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_CHANNEL_CONNECTED at slot
1 and port 0
*Apr  7 12:39:47.191: CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_DSX0_CONNECTED at slot 1,
port 0
*Apr  7 12:39:47.191: Mica Modem(1/0): Link Initiate
*Apr  7 12:39:47.227: Mica Modem(1/3): State Transition to Connect
*Apr  7 12:39:47.287: Mica Modem(1/0): State Transition to Connect
*Apr  7 12:39:49.103: Mica Modem(1/0): State Transition to Link
*Apr  7 12:39:52.103: Mica Modem(1/3): State Transition to Link
*Apr  7 12:40:00.927: Mica Modem(1/3): State Transition to Trainup
*Apr  7 12:40:00.991: Mica Modem(1/0): State Transition to Trainup
*Apr  7 12:40:02.615: Mica Modem(1/0): State Transition to EC Negotiating
*Apr  7 12:40:02.615: Mica Modem(1/3): State Transition to EC Negotiating
CONNECT 31200 /V.42/V.42bis
Router>
*Apr  7 12:40:05.983: Mica Modem(1/0): State Transition to Steady State
*Apr  7 12:40:05.983: Mica Modem(1/3): State Transition to Steady State+++
OK
ath
*Apr  7 12:40:09.167: Mica Modem(1/0): State Transition to Steady State Escape
*Apr  7 12:40:10.795: Mica Modem(1/0): State Transition to Terminating
*Apr  7 12:40:10.795: Mica Modem(1/3): State Transition to Terminating
*Apr  7 12:40:11.755: Mica Modem(1/3): State Transition to Idle
*Apr  7 12:40:11.755: Mica Modem(1/3): Went onhook
*Apr  7 12:40:11.755: CSM_PROC_IC5_OC6_CONNECTED: CSM_EVENT_MODEM_ONHOOK at slot 1, port 3
*Apr  7 12:40:11.755: VDEV_DEALLOCATE: slot 1 and port 3 is deallocated
*Apr  7 12:40:11.759: neat msg at slot 0: (0/0): Tx LOOP_OPEN (ABCD=0101)
*Apr 7 12:40:11.767: neat msg at slot 0: (1/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:40:12.087: neat msg at slot 0: (1/0): Tx LOOP_OPEN (ABCD=0101)
*Apr  7 12:40:12.091: neat msg at slot 0: (0/0): Rx LOOP_OPEN (ABCD=0101)
*Apr  7 12:40:12.111: CSM_RX_CAS_EVENT_FROM_NEAT:(A001):  EVENT_CALL_IDLE at slot 1 and
port 0
```

```
*Apr  7 12:40:12.111: CSM_PROC_IC5_OC6_CONNECTED: CSM_EVENT_DSX0_DISCONNECTED at slot 1,
port 0
*Apr  7 12:40:12.111: Mica Modem(1/0): Link Terminate(0x6)
*Apr  7 12:40:12.779: Mica Modem(1/3): State Transition to Terminating
*Apr  7 12:40:12.839: Mica Modem(1/3): State Transition to Idle
*Apr  7 12:40:13.495: Mica Modem(1/0): State Transition to Idle
*Apr  7 12:40:13.495: Mica Modem(1/0): Went onhook
*Apr  7 12:40:13.495: CSM_PROC_IC6_OC8_DISCONNECTING: CSM_EVENT_MODEM_ONHOOK at slot 1,
port 0
*Apr  7 12:40:13.495: VDEV_DEALLOCATE: slot 1 and port 0 is deallocated
Router#disc
Closing connection to 1.19.36.7 [confirm]
Router#
*Apr  7 12:40:18.783: Mica Modem(1/0): State Transition to Terminating
*Apr  7 12:40:18.843: Mica Modem(1/0): State Transition to Idle
Router#
```

The MICA technologies modem goes through the following internal link states when the call comes in:

- Call Setup
- Off Hook
- Connect
- Link
- Trainup
- EC Negotiation
- Steady State

The following section describes the CSM activity for an incoming call.

When a voice call comes in, CSM is informed of the incoming call. This allocates the modem and sends the Call Setup message to the MICA modem. The Call_Proc message is sent through D channel. The modem sends an offhook message to CSM by sending the state change to Call Setup. The D channel then sends a CONNECT message. When the CONNECT_ACK message is received, the Link initiate message is sent to the MICA modem and it negotiates the connection with the remote modem. In the following debug examples, a modem on slot 1, port 13 is allocated. It goes through its internal states before it is in Steady State and answers the call.

```
Router# debug modem csm
Modem Management Call Switching Module debugging is on
*May 13 15:01:00.609: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xE, ces=0x1
   bchan=0x12, event=0x1, cause=0x0
*May 13 15:01:00.609: VDEV_ALLOCATE: slot 1 and port 13 is allocated.
*May 13 15:01:00.609: MODEM_REPORT(000E): DEV_INCALL at slot 1 and port 13
*May 13 15:01:00.609: CSM_PROC_IDLE: CSM_EVENT_ISDN_CALL at slot 1, port 13
*May 13 15:01:00.609: Mica Modem(1/13): Configure(0x0)
*May 13 15:01:00.609: Mica Modem(1/13): Configure(0x0)
*May 13 15:01:00.609: Mica Modem(1/13): Configure(0x6)
*May 13 15:01:00.609: Mica Modem(1/13): Call Setup
*May 13 15:01:00.661: Mica Modem(1/13): State Transition to Call Setup
*May 13 15:01:00.661: Mica Modem(1/13): Went offhook
*May 13 15:01:00.661: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 13
*May 13 15:01:00.661: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xE, ces=0x1
   bchan=0x12, event=0x4, cause=0x0
*May 13 15:01:00.661: MODEM_REPORT(000E): DEV_CONNECTED at slot 1 and port 13
```

```
*May 13 15:01:00.665: CSM_PROC_IC3_WAIT_FOR_CARRIER:
CSM_EVENT_ISDN_CONNECTED at slot 1, port 13
*May 13 15:01:00.665: Mica Modem(1/13): Link Initiate
*May 13 15:01:00.693: Mica Modem(1/13): State Transition to Connect
*May 13 15:01:01.109: Mica Modem(1/13): State Transition to Link
*May 13 15:01:09.433: Mica Modem(1/13): State Transition to Trainup
*May 13 15:01:11.541: Mica Modem(1/13): State Transition to EC Negotiating
*May 13 15:01:12.501: Mica Modem(1/13): State Transition to Steady State
```

The following section describes the status of CSM when a call is connected.

The **show modem csm x/y** command is similar to AS5200 access server. For an active incoming analog call, the modem_status and csm_status should be VDEV_STATUS_ACTIVE_CALL and CSM_IC4_CONNECTED, respectively.

```
Router# show modem csm 1/13
MODEM_INFO: slot 1, port 13, unit 0, modem_mask=0x0000, modem_port_offset=0
tty_hwidb=0x60D0BCE0, modem_tty=0x60B6FE7C, oobp_info=0x00000000,
modem_pool=0x60ADC998
modem_status(0x0002):VDEV_STATUS_ACTIVE_CALL.
csm_state(0x0204)=CSM_IC4_CONNECTED, csm_event_proc=0x600C6968, current
call thru PRI line
invalid_event_count=0, wdt_timeout_count=0
wdt_timestamp_started is not activated
wait_for_dialing:False, wait_for_bchan:False
pri_chnl=TDM_PRI_STREAM(s0, u0, c18), modem_chnl=TDM_MODEM_STREAM(s1, c13)
dchan_idb_start_index=0, dchan_idb_index=0, call_id=0x000E, bchan_num=18
csm_event=CSM_EVENT_ISDN_CONNECTED, cause=0x0000
ring_indicator=0, oh_state=0, oh_int_enable=0, modem_reset_reg=0
ring_no_answer=0, ic_failure=0, ic_complete=1
dial_failure=0, oc_failure=0, oc_complete=0
oc_busy=0, oc_no_dial_tone=0, oc_dial_timeout=0
remote_link_disc=0, stat_busyout=0, stat_modem_reset=0
oobp_failure=0
call_duration_started=1d02h, call_duration_ended=00:00:00,
total_call_duration=00:00:00
The calling party phone number = 4085552400
The called party phone number  = 4085551400
total_free_rbs_timeslot = 0, total_busy_rbs_timeslot = 0,
total_dynamic_busy_rbs_timeslot = 0, total_static_busy_rbs_timeslot = 0,
min_free_modem_threshold = 6
```

The following section describes the CSM activity for an outgoing call.

For MICA modems, the dial tone is not required to initiate an outbound call. Unlike in the AS5200, the digit collection step is not required. The dialed digit string is sent to the CSM in the outgoing request to the CSM. CSM signals the D channel to generate an outbound voice call, and the B channel assigned is connected to the modem and the CSM.

The modem is ordered to connect to the remote side with a CONNECT message, and by sending a link initiate message, the modem starts to train.

```
Router# debug modem csm
Modem Management Call Switching Module debugging is on
Router# debug isdn q931
ISDN Q931 packets debugging is on
*May 15 12:48:42.377: Mica Modem(1/0): Rcvd Dial String(5552400)
*May 15 12:48:42.377: CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
*May 15 12:48:42.377: CSM_PROC_OC3_COLLECT_ALL_DIGIT:
CSM_EVENT_GET_ALL_DIGITS at slot 1, port 0
*May 15 12:48:42.377: CSM_PROC_OC3_COLLECT_ALL_DIGIT: called party num:
(5552400) at slot 1, port 0
```

```
*May 15 12:48:42.381:  process_pri_call making a voice_call.
*May 15 12:48:42.381: ISDN Se0:23: TX ->  SETUP pd = 8  callref = 0x0011
*May 15 12:48:42.381:         Bearer Capability i = 0x8090A2
*May 15 12:48:42.381:         Channel ID i = 0xE1808397
*May 15 12:48:42.381:         Called Party Number i = 0xA1, '5552400'
*May 15 12:48:42.429: ISDN Se0:23: RX <-  CALL_PROC pd = 8  callref = 0x8011
*May 15 12:48:42.429:         Channel ID i = 0xA98397
*May 15 12:48:42.429: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xA011, ces=0x1
   bchan=0x16, event=0x3, cause=0x0
*May 15 12:48:42.429: MODEM_REPORT(A011): DEV_CALL_PROC at slot 1 and port 0
*May 15 12:48:42.429: CSM_PROC_OC4_DIALING: CSM_EVENT_ISDN_BCHAN_ASSIGNED
at slot 1, port 0
*May 15 12:48:42.429: Mica Modem(1/0): Configure(0x1)
*May 15 12:48:42.429: Mica Modem(1/0): Configure(0x0)
*May 15 12:48:42.429: Mica Modem(1/0): Configure(0x6)
*May 15 12:48:42.429: Mica Modem(1/0): Call Setup
*May 15 12:48:42.489: Mica Modem(1/0): State Transition to Call Setup
*May 15 12:48:42.589: ISDN Se0:23: RX <-  ALERTING pd = 8  callref = 0x8011
*May 15 12:48:43.337: ISDN Se0:23: RX <-  CONNECT pd = 8  callref = 0x8011
*May 15 12:48:43.341: MODEM_REPORT:dchan_idb=0x60D437F8, call_id=0xA011, ces=0x1
   bchan=0x16, event=0x4, cause=0x0
*May 15 12:48:43.341: MODEM_REPORT(A011): DEV_CONNECTED at slot 1 and port 0
*May 15 12:48:43.341: CSM_PROC_OC5_WAIT_FOR_CARRIER:
CSM_EVENT_ISDN_CONNECTED at slot 1, port 0
*May 15 12:48:43.341: Mica Modem(1/0): Link Initiate
*May 15 12:48:43.341: ISDN Se0:23: TX ->  CONNECT_ACK pd = 8  callref = 0x0011
*May 15 12:48:43.385: Mica Modem(1/0): State Transition to Connect
*May 15 12:48:43.849: Mica Modem(1/0): State Transition to Link
*May 15 12:48:52.665: Mica Modem(1/0): State Transition to Trainup
*May 15 12:48:54.661: Mica Modem(1/0): State Transition to EC Negotiating
*May 15 12:48:54.917: Mica Modem(1/0): State Transition to Steady State
```

Related Commands

Command	Description
debug modem oob	Creates modem startup messages between the network management software and the modem on the specified OOB port.
debug modem trace	Performs a call trace on the specified modem, which allows you to determine why calls are terminated.

debug modem dsip

To display output for modem control messages that are received or sent to the router, use the **debugmodemdsip** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem dsip {*tty-range* | **group** | *shelf/slot/port*}
no debug modem dsip {*tty-range* | **group** | *shelf/slot/port*}

Syntax Description

tty-range	Modem tty number or range. You can specify a single TTY line number or a range from 0 through the number of modems you have in your Cisco AS5800 access server. Be sure to include a dash (-) between the range values you specify.
group	Modem group information.
shelf/slot/port	Location of the modem by shelf/slot/port numbers for internal modems.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(2)AA	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The**debugmodemdsip**command displays each Distributed System Interconnect Protocol (DSIP) message that relates to a modem and is sent from or received at the router shelf. This command can be applied to a single modem or a group of modems.

Examples

The following examples show a display of the available **debugmodem** command options and **debugmodemdsip** command options:

```
Router# debug modem ?
  dsip         Modem DSIP activity
  maintenance  Modem maintenance activity
  oob          Modem out of band activity
  trace        Call Trace Upload
  traffic      Modem data traffic
  <cr>

Router# debug modem dsip ?
  <0-935>  First Modem TTY Number
  group    Modem group information
  x/y/z    Shelf/Slot/Port for Internal Modems
  <cr>
```

The following example indicates that a Real Time Server (RTS) status message was received from the router shelf, and an ACK message was sent back:

```
Router# debug modem dsip
00:11:02: RSMODEM_SEND-1/2/06: MODEM_RING_INDICATION_MSG cci1 si0 ms0 mm65535,0 dc0
00:11:02: RSMODEM_sRCV-1/2/06:l12,MODEM_CALL_ACK_MSG:
```

```
00:11:02: RSMODEM_SEND-1/2/06: MODEM_CALL_ACCEPT_MSG
00:11:11: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 0 36 21
00:11:18: RSMODEM_sRCV-1/2/06:l12,MODEM_SET_DCD_STATE_MSG: 1
00:11:19: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:11:19: RSMODEM_dRCV-2:l1258607996,MODEM_RTS_STATUS_MSG: 0 6 0 23 0 0 0 0 0
00:11:23: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 0 150 21
00:12:31: RSMODEM_sRCV-1/2/06:l12,MODEM_SET_DCD_STATE_MSG: 0
00:12:31: RSMODEM_SEND-1/2/06: MODEM_CALL_HANGUP_MSG
00:12:31: RSMODEM_sRCV-1/2/06:l12,MODEM_ONHOOK_MSG:
00:12:32: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:12:32: RSMODEM_SEND-1/2/06: MODEM_SET_DTR_STATE_MSG 0
00:12:32: RSMODEM_dRCV-2:l1258659676,MODEM_RTS_STATUS_MSG: 0 6 0 16 0 0 0 0 0
00:12:32: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:12:32: RSMODEM_dRCV-2:l1258600700,MODEM_RTS_STATUS_MSG: 0 6 0 13 0 0 0 0 0
00:12:33: RSMODEM_SEND-1/2/06: MODEM_SET_DTR_STATE_MSG 0
00:12:33: RSMODEM_SEND-1/2/06: MODEM_RTS_STATUS_MSG 1
00:12:33: RSMODEM_dRCV-2:l1258662108,MODEM_RTS_STATUS_MSG: 0 6 0 16 0 0 0 0 0
00:12:35: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 1 34 22
00:12:38: RSMODEM_SEND-1/2/06: MODEM_SET_DTR_STATE_MSG 1
00:12:47: RSMODEM_sRCV-2:l0,MODEM_POLL_MSG: 0 16 0 7 0 146 0 12 22
```

The following table describes the significant fields shown in the display.

Table 1: debug modem dsip Field Descriptions

Field	Description
RSMODEM_SEND-1/2/06	Router shelf modem shelf sends a MODEM_RING_INDICATION_MSG message.
RSMODEM_sRCV-1/2/06	Router shelf modem received a MODEM_CALL_ACK_MSG message.
MODEM_CALL_ACCEPT_MSG	Router shelf accepts the call.
MODEM_CALL_HANGUP_MSG	Router shelf sends a hangup message.
MODEM_RTS_STATUS_MSG	Request to send message status.

Related Commands

Command	Description
debug dsip	Displays output for DSIP used between the router shelf and the dial shelf.
debug modem traffic	Displays output for framed, unframed, and asynchronous data transmission received from the modem cards.

debug modem oob

To debug the out-of-band port used to poll modem events on the modem, use the **debug modem oob**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem oob [{*slot/modem-port* | **group** *group-number*}]
no debug modem oob [{*slot/modem-port* | **group** *group-number*}]

Syntax Description

slot/modem-port	(Optional) The slot and modem port number.
group *group-number*	(Optional) The modem group.

Command Modes

Privileged EXEC

Usage Guidelines

The message types and sequence numbers that appear in the debugging output are initiated by the Modem Out-of-Band Protocol and used by service personnel for debugging purposes.

Caution Entering the **debug modem oob** command without specifying a slot and modem number debugs *all* out-of-band ports, which generates a substantial amount of information.

Examples

The following is sample output from the **debug modem oob**command. This example debugs the out-of-band port on modem 2/0, which creates modem startup messages between the network management software and the modem.

```
Router# debug modem oob 2/0
MODEM(2/0): One message sent --Message type:3, Sequence number:0
MODEM(2/0): Modem DC session data reply
MODEM(2/0): One message sent --Message type:83, Sequence number:1
MODEM(2/0): DC session event =
MODEM(2/0): One message sent --Message type:82, Sequence number:2
MODEM(2/0): No status changes since last polled
MODEM(2/0): One message sent --Message type:3, Sequence number:3
MODEM(2/0): Modem DC session data reply
MODEM(2/0): One message sent --Message type:83, Sequence number:4
```

debug modem relay errors

To view modem relay network errors, use the **debug modem relay errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **errors**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **errors**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers and the Cisco AS5300 universal access server.

Usage Guidelines

In a stable modem relay network, the **debug modem relay errors** command produces little output.

Examples

The following is sample output from the **debug modem relay errors** command. The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 28 tm 11944 OUT ERR, pb=12, payload: 00 06
 00 00 00 00 00 07 00 00 01 DE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 29 tm 11944 OUT ERR, pb=12, payload: 00
06 00 00 00 00 00 04 00 00 00 BE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 30 tm 11944 OUT ERR, pb=12, payload: 00
06 00 00 00 00 00 05 FF FF FF FD
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay events	Displays events that may cause failure of the modem relay network.

debug modem relay events

To view the events that may cause failure of the modem relay network, use the **debug modem relay events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **events**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **events**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers and the Cisco AS5300 universal access server.

Usage Guidelines

In a stable modem relay network, the **debug modem relay events**command produces little output.

Examples

The following is sample output from the **debug modem relay events** command. The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Router# debug modem relay events
Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 28 tm 11944 OUT EVNT, pb=12, payload: 00
06 00 00 00 00 00 07 00 00 01 DE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 29 tm 11944 OUT EVNT, pb=12, payload: 00
06 00 00 00 00 00 04 00 00 00 BE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 30 tm 11944 OUT EVNT, pb=12, payload: 00
06 00 00 00 00 00 05 FF FF FF FD
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay packetizer

To view events occurring in the modem relay packetizer module, use the **debug modem relay packetizer** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **packetizer**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **packetizer**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay packetizer**command. Using the **debug modem relay packetizer**command generates a large volume of debugs, which can affect router performance.

Examples

The following is sample output from the **debug modem relay packetizer** command. The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Router# debug modem relay packetizer

Jan 11 05:33:33.715:ModemRelay pkt[0:D:11]. sqn 8 tm 47610 IN PKTZR, pb=7, payload: 82 38
00 18 03 01 87
*Jan 11 05:33:33.727:ModemRelay pkt[0:D:11]. sqn 9 tm 47616 OUT PKTZR, pb=7, payload: 82
20 00 18 03 01 47
*Jan 11 05:33:35.719:ModemRelay pkt[0:D:11]. sqn 10 tm 49614 IN PKTZR, pb=7, payload: 82
39 00 18 03 01 87
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay physical

To view modem relay physical layer packets, use the **debug modem relay physical** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **physical**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **physical**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay physical**command. Using the **debug modem relay physical**command generates a large volume of debugs, which can affect router performance.

Examples

The following is sample output from the **debug modem relay physical** command. The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 28 tm 11944 OUT PHYS, pb=12, payload: 00
06 00 00 00 00 00 07 00 00 01 DE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 29 tm 11944 OUT PHYS, pb=12, payload: 00
06 00 00 00 00 00 04 00 00 00 BE
*Jan 11 05:35:09.119:ModemRelay pkt[0:D:11]. sqn 30 tm 11944 OUT PHYS, pb=12, payload: 00
06 00 00 00 00 00 05 FF FF FF FD
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay sprt

To view modem relay Simple Packet Relay Transport (SPRT) protocol packets, use the **debug modem relay sprt** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **sprt**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **sprt**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay sprt** command. Using the **debug modem relay sprt** command generates a large volume of debugs, which can affect router performance.

Examples

The following is sample output from the **debug modem relay sprt** command. The output shows the sequence number of the packet, time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:37:16.151:ModemRelay pkt[0:D:11]. sqn 34 tm 7910 OUT SPRT, pb=4, payload: 02 00
03 71
*Jan 11 05:37:16.295:ModemRelay pkt[0:D:11]. sqn 35 tm 8048 IN SPRT, pb=13, payload: 02 00
 01 F1 F7 7E FD F5 90 F3 3E 90 55
*Jan 11 05:37:16.303:ModemRelay pkt[0:D:11]. sqn 36 tm 8060 IN SPRT, pb=6, payload: 02 00
01 41 04 00
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay udp

To view events occurring in the User Datagram Protocol (UDP) stack, use the **debug modem relay udp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **udp**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **udp**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay udp** command. Using the **debug modem relay udp** command generates a large volume of debugs, which can affect router performance.

Examples

The following is sample output from the **debug modem relay udp** command. The output shows three UDP packets related to modem relay. In the sample output, OUT or IN represent packet direction, and UDP indicates the specific layer that reported the packet.

```
Jan 1 03:39:29.407:ModemRelay pkt[0:D (4)]. sqn 61 tm 3060 OUT UDP, pb=6, payload: 80 00
00 00 00 00
*Jan 1 03:39:29.471:ModemRelay pkt[0:D (4)]. sqn 62 tm 3120 IN UDP, pb=6, payload: 40 00
00 00 00 00
*Jan 1 03:39:29.471:ModemRelay pkt[0:D (4)]. sqn 63 tm 3120 IN UDP, pb=6, payload: 80 00
00 00 00 00
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem relay v14

To observe events occurring in the V.14 layer, use the **debug modem relay v14** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **v14**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **v14**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

No debugging output is displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(2)T	This command was introduced .

Usage Guidelines

Use the **debug modem relay v14**command to debug V.14 layer modem relay calls. Using this command generates a large volume of debugs, which can affect router performance; disable console logging and use buffered logging before using the **debug modem relay v14**command. In most instances you will use this command only at the request of Cisco Technical Assistance Center (TAC).

Examples

The following is sample output from the **debug modem relay v14** command. The output shows the sequence number of the packet time stamp, direction, layer, and payload bytes, followed by each byte of the payload in hexadecimal.

```
Router# debug modem relay v14
*Aug 10 22:51:37.496: ModemRelay pkt[1/1:1]. sqn 15649 tm 48766 OUT V14, pb=18, payload:
08 BC 4C 51 CE 1A 69 ED D6 65 62 8C 7F D3 9A 82 5A 7A
*Aug 10 22:51:38.216: ModemRelay pkt[1/1:1]. sqn 15650 tm 48778 IN V14, pb=22, payload:
9A 9C 7F 57 2D D7 4C 98 E8 EC FC 73 69 F2 FF A3 E8 B0 A4 58 BB AE
*Aug 10 22:51:38.216: ModemRelay pkt[1/1:1]. sqn 15651 tm 48790 OUT V14, pb=18, payload:
64 F9 73 D3 AB 11 61 ED 1E 5D 51 8D B1 9F CA 49 BF F4
*Aug 10 22:51:38.216: ModemRelay pkt[1/1:1]. sqn 15652 tm 48796 IN V14, pb=21, payload:
C1 77 90 12 F8 37 E8 7A 64 8D 0E 61 58 7E E4 E8 87 E0 B4 83 C7 A4 60 7A 64 8B 09 B9 80 2E
E5 2E 94 65 79 C2 A8 E9 6F D9 6C 3B
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.

Command	Description
debug modem relay errors	Displays modem relay network errors.

debug modem relay v42

To view events occurring in the V.42 layer, use the **debug modem relay v42** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem relay [**call-identifier** *call-setup-time call-index*] **v42**
no debug modem relay [**call-identifier** *call-setup-time call-index*] **v42**

Syntax Description

call-identifier	(Optional) Identifies a particular call.
call-setup-time	(Optional) Value of the system UpTime when the call associated with this entry was started. Valid values are 0 through 4294967295.
call-index	(Optional) Dial peer identification number used to distinguish between calls with the same setup time. Valid values are 0 through 10.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced for the Cisco 2600, Cisco 3620, Cisco 3640, Cisco 3660, and Cisco 7200 series routers, and the Cisco AS5300 universal access server.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Disable console logging and use buffered logging before using the **debug modem relay v42** command. Using the **debug modem relay v42** command generates a large volume of debugs, which can affect router performance.

Examples

The following is sample output from the **debug modem relay v42** command. The output shows the sequence number of the packet, timestamp, direction, layer, and payload-bytes, followed by each byte of the payload in hexadecimal.

```
Jan 11 05:42:08.715:ModemRelay pkt[0:D:13]. sqn 3 tm 10104 OUT V42, pb=43, payload: 03 AF
82 80 00 13 03 03 8A 89 00 05 02 03 E0 06 02 03 E0 07 01 08 08 01 08 F0 00 0F 00 03 56 34
32 01 01 03 02 02 04 00 03 01 20
*Jan 11 05:42:08.847:ModemRelay pkt[0:D:13]. sqn 4 tm 10236 IN V42, pb=2, payload: 03 7F
```

Related Commands

Command	Description
debug hpi all	Displays gateway DSP modem relay termination codes.
debug modem relay errors	Displays modem relay network errors.

debug modem trace

To debug a call trace on the modem to determine why calls are terminated, use the **debug modem trace**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem trace [{**normal** | **abnormal** | **all**}] [{*slot*/*modem-port* | **group** *group-number*}]
no debug modem trace [{**normal** | **abnormal** | **all**}] [{*slot*/*modem-port* | **group** *group-number*}]

Syntax Description

normal	(Optional) Uploads the call trace to the syslog server on normal call termination (for example, a local user hangup or a remote user hangup).
abnormal	(Optional) Uploads the call trace to the syslog server on abnormal call termination (for example, any call termination other than normal termination, such as a lost carrier or a watchdog timeout).
all	(Optional) Uploads the call trace on all call terminations including normal and abnormal call termination.
slot/modem-port	(Optional) The slot and modem port number.
group *group-number*	(Optional) The modem group.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug modem trace** command applies only to manageable modems. For additional information, use the **show modem**command.

Examples

The following is sample output from the **debug modem trace abnormal** command:

```
Router# debug modem trace abnormal 1/14
Modem 1/14 Abnormal End of Connection Trace. Caller 123-4567
    Start-up Response: AS5200 Modem, Firmware 1.0
    Control Reply: 0x7C01
    DC session response: brasil firmware 1.0
    RS232 event:
    DSR=On, DCD=On, RI=Off, TST=Off
    changes: RTS=No change, DTR=No change, CTS=No change
    changes: DSR=No change, DCD=No change, RI=No change, TST=No change
    Modem State event: Connected
    Connection event: Speed = 19200, Modulation = VFC
    Direction = Originate, Protocol = reliable/LAPM, Compression = V42bis
    DTR event: DTR On
    Modem Activity event: Data Active
    Modem Analog signal event: TX = -10, RX = -24, Signal to noise = -32
    End connection event: Duration = 10:34-11:43,
    Number of xmit char =    67, Number of rcvd char = 88, Reason: Watchdog Time-out.
```

Related Commands

Command	Description
debug modem csm	Debugs the CSM used to connect calls on the modem.

Command	Description
debug modem oob	Creates modem startup messages between the network management software and the modem on the specified OOB port.

debug modem traffic

To display output for framed, unframed, and asynchronous data sent received from the modem cards, use the **debug modem traffic** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug modem traffic
no debug modem traffic

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(2)AA	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **debug modem traffic** command displays output for framed, unframed, and asynchronous data sent or received by the modem cards.

Examples

The following example displays information about unframed or framed data sent to or received from the modem cards:

```
Router# debug modem traffic
MODEM-RAW-TX:modem = 6/5/00, length = 1, data = 0x61, 0xFF, 0x7D, 0x23
MODEM-RAW-RX:modem = 6/5/00, length = 1, data = 0x61, 0x0, 0x0, 0x0
```

The information indicates unframed asynchronous data transmission and reception involving the modem on shelf 6, slot 5, port 00.

The following example displays framed asynchronous data transmission and reception involving the modem on shelf 6, slot 5, port 00:

```
Router# debug modem traffic
MODEM-FRAMED-TX:modem = 6/5/00, length = 8, data = 0xFF, 0x3, 0x82
MODEM-FRAMED-RX:modem = 6/5/00, length = 14, data = 0xFF, 0x3, 0x80
```

Related Commands

Command	Description
debug modem dsip	Displays output for modem control messages that are received or sent to the router.

debug mpls adjacency

To display changes to label switching entries in the adjacency database, use the **debugmplsadjacency**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls adjacency
no debug mpls adjacency

Usage Guidelines

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI command syntax.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **debugmplsadjacency** command to monitor when entries are updated in or added to the adjacency database.

Examples

The following is sample output from the **debugmplsadjacency** command:

```
Router# debug mpls adjacency
TAG ADJ: add 10.10.0.1, Ethernet0/0/0
TAG ADJ: update 10.10.0.1, Ethernet0/0/0
```

The following table describes the significant fields shown in the sample display.

Table 2: debug mpls adjacency Field Description

Field	Description
add	Adding an entry to the database.
update	Updating the MAC address for an existing entry.
10.10.0.1	Address of neighbor TSR.
Ethernet0/0/0	Connecting interface.

debug mpls atm-cos

Note Effective with Cisco IOS Release 12.4(20)T, the **debug mpls atm-cos** command is not available in Cisco IOS software.

To display ATM label virtual circuit (VC) bind or request activity that is based on the configuration of a Quality of Service (QoS) map, use the **debug mpls atm-cos** command in privileged EXEC mode. To disable this feature, use the **no** form of this command.

debug mpls atm-cos [{**bind** | **request**}]
no debug mpls atm-cos [{**bind** | **request**}]

Syntax Description

bind	(Optional) Specifies debug information about bind responses for a VC path.
request	(Optional) Specifies debug information about bind requests for a VC path.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF syntax and terminology.
12.2(2)T	This command was incorporated into Cisco IOS Release 12.2(2)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was removed.

Examples

The following command sequence demonstrates how to obtain sample output from the **debug mpls atm-cos** command.

First, display the Multiprotocol Label Switching (MPLS) forwarding table to see which prefixes are associated with a single label VC (LVC), as shown below:

```
Router# show mpls forwarding
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
26     28          10.17.17.17/32   0          PO6/0      point2point
27     Pop tag     10.11.11.11/32   1560       PO6/0      point2point
28     27          10.16.16.16/32   0          PO6/0      point2point
29     30          10.92.0.0/8      0          PO6/0      point2point
30     Pop tag     10.95.0.0/8      2600       PO6/0      point2point
31     2/34        10.10.10.10/32   0          AT2/0.1    point2point
32     Pop tag     10.14.14.14/32   0          Fa5/0      10.91.0.1
33     Pop tag     10.90.0.0/8      0          Fa5/0      10.91.0.1
```

```
34      Pop tag     10.96.0.0/8       0          Fa5/0      10.91.0.1
        2/36        10.96.0.0/8       0          AT2/0.1    point2point
35      35          10.93.0.0/8       0          PO6/0      point2point
36      36          10.12.12.12/32    0          PO6/0      point2point
37      37          10.15.15.15/32    0          PO6/0      point2point
38      37          10.18.18.18/32    0          Fa5/0      10.91.0.1
39      39          10.97.0.0/8       540        PO6/0      point2point
40      40          10.98.0.0/8       0          PO6/0      point2point
```

Second, enable debugging of request and bind events, as shown in the command sequence below:

```
Router# debug mpls atm-cos ?
  bind     Bind response for VC path
  request  Requests for VC binds path
Router# debug mpls atm-cos request
ATM TAGCOS VC requests debugging is on
Router# debug mpls atm-cos bind
ATM TAGCOS Bind response debugging is on
```

Third, configure an MPLS ATM subinterface for multi-VC mode. The corresponding request and bind events are displayed, as shown below:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface a2/0.1
Router(config-subif)# mpls atm multi-vc
Router(config-subif)# end
Router#
19:59:14:%SYS-5-CONFIG_I:Configured from console by console
Router#
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, available
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, standard
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, premium
19:59:24:TAGCOS-REQ:vc request 10.10.10.10/32, control
19:59:24:TAGCOS-REQ:vc request 10.96.0.0/8, available
19:59:24:TAGCOS-REQ:vc request 10.96.0.0/8, standard
19:59:24:TAGCOS-REQ:vc request 10.96.0.0/8, premium
19:59:24:TAGCOS-REQ:vc request 10.96.0.0/8, control
TAGCOS-REQ/TCATM:10.11.11.11/32,len=4352,band=1099528405504,class=0x700
TAGCOS-REQ/TCATM:10.12.12.12/32,len=4352,band=2199040033280,class=0x700
TAGCOS-REQ/TCATM:10.13.13.13/32,len=4352,band=3298551661056,class=0x700
TAGCOS-REQ/TCATM:10.14.14.14/32,len=4352,band=4398063288832,class=0x700
TAGCOS-REQ/TCATM:10.15.15.15/32,len=4352,band=5497574916608,class=0x700
TAGCOS-REQ/TCATM:10.16.16.16/32,len=4352,band=6597086544384,class=0x700
TAGCOS-REQ/TCATM:10.17.17.17/32,len=4352,band=7696598172160,class=0x700
TAGCOS-REQ/TCATM:10.18.18.18/32,len=4352,band=8796109799936,class=0x700
TAGCOS-REQ/TCATM:10.90.0.0/8,len=768,band=3940649674539009,class=0x2
TAGCOS-REQ/TCATM:10.91.0.0/8,len=768,band=3940649674604545,class=0x2
TAGCOS-REQ/TCATM:10.92.0.0/8,len=768,band=3940649674670081,class=0x2
TAGCOS-REQ/TCATM:10.93.0.0/8,len=768,band=3940649674735617,class=0x2
TAGCOS-REQ/TCATM:10.94.0.0/8,len=768,band=3940649674801153,class=0x2
TAGCOS-REQ/TCATM:10.95.0.0/8,len=768,band=3940649674866689,class=0x2
TAGCOS-REQ/TCATM:10.97.0.0/8,len=768,band=3940649674932225,class=0x2
TAGCOS-REQ/TCATM:10.98.0.0/8,len=768,band=3940649674997761,class=0x2
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=41 - control 41,41,41,41
TAGCOS-BIND:binding_ok 10.10.10.10/32, Inform TFIB pidx=0, in_tag=31, idx=0x80000000
TAGCOS-BIND:binding_ok 10.96.0.0/8,VCD=42 - control 42,42,42,42
TAGCOS-BIND:binding_ok 10.96.0.0/8, Inform TFIB pidx=1, in_tag=34, idx=0x80000001
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=43 - premium 43,43,43,41
TAGCOS-BIND:binding_ok 10.96.0.0/8,VCD=44 - premium 44,44,44,42
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=45 - standard 45,45,43,41
TAGCOS-BIND:binding_ok 10.96.0.0/8,VCD=46 - standard 46,46,44,42
```

```
TAGCOS-BIND:binding_ok 10.10.10.10/32,VCD=47 - available 47,45,43,41
TAGCOS-BIND:binding_ok 10.96.0.0/8,VCD=48 - available 48,46,44,42
```

debug mpls atm-ldp api

Note Effective with Cisco IOS Release 12.4(20)T, the **debugmplsatm-ldpapi** command is not available in Cisco IOS software.

To display information about the virtual channel identifier (VCI) allocation of label virtual circuits (LVCs), label-free requests, and cross-connect requests, use the **debugmplsatm-ldpapi**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls atm-ldp api
no debug mpls atm-ldp api

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was modified.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was removed.

Usage Guidelines

Use the **debugmplsatm-ldpapi** command in conjunction with the**debugmplsatm-ldproutes**and **debugmplsatm-ldpstates**command to display more complete information about an LVC.

Examples

The following shows sample output from the **debugmplsatm-ldpapi** command:

```
Router# debug mpls atm-ldp api
Tailend Router Free label Req 167.50.0.0 on ATM0/0.2 VPI/VCI 1/674
 TAGATM_API: received label free request
             interface: ATM0/0.2  dir: in  vpi: 1  vci: 674
 TAGATM_API: completed label free
             interface: ATM0/0.2  vpi: 1  vci: 674
             result: TAGATM_OK
```

The following table describes the significant fields shown in the display.

Table 3: debug mpls atm-ldp api Field Descriptions

Field	Description
TAGATM_API	Subsystem that displays the message.
interface	Interface used by the driver to allocate or free VPI/VCI resources.
dir	Direction of the VC: • In--Input or receive VC • Out--Output VC
vpi	Virtual path identifier.
vci	Virtual channel identifier.
result	The return error code from the driver API.

Related Commands

Command	Description
debug mpls atm-ldp states	Displays information about LVC state transitions as they occur.

debug mpls atm-ldp failure

To display failure information about the LC-ATM, use the **debug mpls atm-ldp failure** command in privileged EXEC mode. To disable this feature, use the no form of the command.

debug mpls atm-ldp failure
no debug mpls atm-ldp failure

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(8)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **debug mpls atm-ldp failure** command to display failure information about the LC-ATM. This command is useful for determining failure cases. This command displays only failure information, unlike the **debug mpls atm-ldp api** command, which displays all API events.

Examples

This section shows sample output from the **debug mpls atm-ldp failure** command.

The following failure message displays during a race condition where the LC-ATM attempts to allocate label virtual circuits (LVCs) on an interface where MPLS has been disabled:

```
Router# debug mpls atm-ldp failure
TAGATM_API_FAILURE: allocate_tag_req on ATM1/0/0 tagsw not enabled
```

The following failure message displays when the LC-ATM fails to deallocate the output leg LVC of a cross connect:

```
Router# debug mpls atm-ldp failure
TAGATM_API_FAILURE: connDeAllocateHalfLeg returned false interface: ATM1/0/0
vpi: 1  vci: 48
```

The following failure message displays when a cross connect cannot be installed on the switching fabric. The result code is also provided.

```
Router# debug mpls atm-ldp failure
TAGATM_API_FAILURE: setup_xconn_req InstallSvcXconn failed result
```

The following message displays when attempts to establish a cross connect fail. The result describes the reason for the failure.

```
Router# debug mpls atm-ldp failure
TCATM-4-XCONNECT_FAILED: 10.254.13.237/32 for ATM0/1/2 ATM1/0/0
TAGATM_API: x-conn setup request completed
             input interface: ATM0/1/2  vpi: 1  vci: 48
            output interface: ATM1/0/0  vpi: 2  vci: 2038
            result = TAGATM_FAIL
Xconnect setup response for 10.254.13.215: failure, 8
```

The following message displays when attempts to remove a cross connect fail. The result describes why the cross connect cannot be removed.

```
Router# debug mpls atm-ldp failure
TCATM-4-XCONNECT_REMOVE_FAILED: Remove XConnect API failed for ATM1/0/12 1/894
-> ATM1/0/13 1/528
TAGATM_API: x-conn remove request completed
             input interface: ATM1/0/12  vpi: 1  vci: 894
            output interface: ATM1/0/13  vpi: 1  vci: 528
            result = TAGATM_FAIL
```

Related Commands

Command	Description
debug mpls atm-ldp api	Displays all driver API events.

debug mpls atm-ldp routes

Note Effective with Cisco IOS Release 12.4(20)T, the **debugmplsatm-ldproutes** command is not available in Cisco IOS software.

To display information about the state of the routes for which virtual circuit identifier (VCI) requests are being made, use the **debugmplsatm-ldproutes**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls atm-ldp routes
no debug mpls atm-ldp routes

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Command	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was modified.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Command	Modification
12.4(20)T	This command was removed.

Usage Guidelines

When there are many routes and system activities (that is, shutting down interfaces, learning new routes, and so forth), the **debugmplsatm-ldproutes** command displays extensive information that might interfere with system timing. Most commonly, this interference affects normal label distribution protocol (LDP) operation. To avoid this problem, you can increase the LDP hold time by means of the **mplsldpholdtime** command.

Examples

The following shows sample output from the**debugmplsatm-ldproutes** command:

```
Router# debug mpls atm-ldp routes
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.7.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.7.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.7.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.8.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.8.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.8.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.9.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.9.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.9.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.10.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.10.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.10.0.0/16
CleanupRoutes,not deleting route of idb ATM0/0.2,rdbIndex 0
tcatmFindRouteTags,153.11.0.0/16,idb=ATM0/0.2,nh=134.111.102.98,index=0
AddNewRoute,153.11.0.0/16,idb=ATM0/0.2
CleanupRoutes,153.11.0.0/16
```

The following table describes the significant fields shown in the display.

Table 4: debug mpls atm-ldp routes Field Descriptions

Field	Description
CleanupRoutes	Cleans up the routing table after a route has been deleted.
not deleting route of idb ATM0/0.2	The route cleanup event has not removed the specified route.
rdbIndex	Index identifying the route.
tcatmFindRouteTags	Request a VC for the route.
idb	The internal descriptor for an interface.
nh	Next hop for the route.
index	Identifier for the route.
AddNewRoute	Action of adding routes for a prefix or address.

Related Commands

Command	Description
mpls ldp holdtime	Changes the time an LDP session is maintained in the absence of LDP messages from the session peer.

debug mpls atm-ldp states

Note Effective with Cisco IOS Release 12.4(20)T, the **debugmplsatm-ldpstates** command is not available in Cisco IOS software.

To display information about label virtual circuit (LVC) state transitions as they occur, use the **debugmplsatm-ldpstates**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls atm-ldp states
no debug mpls atm-ldp states

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was modified.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Release	Modification
12.4(20)T	This command was removed.

Usage Guidelines

When there are many routes and system activities (such as shutting down interfaces, learning new routes, and so forth), the **debugmplsatm-ldpstates** command outputs extensive information that might interfere with system timing. Most commonly, this interference affects normal label distribution protocol (LDP) operation. To avoid this problem, you should increase the LDP hold time by means of the **mplsldpholdtime** command.

Examples

The following shows sample output from the**debugmplsatm-ldpstates**command:

```
Router# debug mpls atm-ldp states
Transit Output 166.35.0.0 VPI/VCI 1/67 Active -> XmitRelease NoPath
Transit Input 166.35.0.0 VPI/VCI 1/466 Active -> ApiWaitParentLoss ParentLoss
Transit Input 166.35.0.0 VPI/VCI 1/466 ApiWaitParentLoss -> ParentWait ApiSuccess
Transit Input 166.35.0.0 VPI/VCI 1/466 ParentWait -> XmitWithdraw NoPath
Transit Input 166.35.0.0 VPI/VCI 1/466 XmitWithdraw -> XmitWithdraw Transmit
Transit Input 166.35.0.0 VPI/VCI 1/466 XmitWithdraw -> NonExistent Release
Transit Input 166.35.0.0 VPI/VCI 1/466 NonExistent -> NonExistent ApiSuccess
```

The following table describes the significant fields shown in the display.

Table 5: debug mpls atm-ldp states Field Descriptions

Field	Description
Transit Output	Output side of an LVC.
VPI/VCI	VC value.
Transit Input	Input side of an LVC.

Related Commands

Command	Description
mpls ldp holdtime	Changes the time an LDP session is maintained in the absence of LDP messages from the session peer.

debug mpls checkpoint label-binding

To display the events for the checkpoint label bindings of Multiprotocol Label Switching (MPLS) applications running on the router, use the debug mpls checkpoint label-binding command in privileged EXEC mode. To disable the display of these events, use the **no** form of this command.

debug mpls checkpoint label-binding
no debug mpls checkpoint label-binding

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series router.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use this command with caution. The command displays the events for every label binding.

Examples

The following example shows the output when you issue the command on the standby Route Processor:

```
Router# debug mpls checkpoint label-binding
MPLS Label Binding Checkpoint debugging is on
3d17h: mpls_lbl_bind_chkpt: client ID 13 up, total client 1
3d17h: mpls_lbl_bind_chkpt: msg rx for 1D, vers 0, type 1
action 56, len 0, state 4, peer 13
3d17h: mpls_lbl_bind_chkpt: post msg type 1
3d17h: mpls_lbl_bind_chkpt: msg rx for 1D, vers 0, type 1
action 56, len 0, state 4, peer 13
3d17h: mpls_lbl_bind_chkpt: post msg type 1
3d17h: mpls_lbl_bind_chkpt: msg rx for 1D, vers 0, type 1
action 56, len 0, state 4, peer 13
3d17h: mpls_lbl_bind_chkpt: post msg type 1
3d17h: mpls_lbl_bind_chkpt: appl_id 13, KEY 000C800018888200
3d17h: mpls_chkpt_db: AVL insert successful, Key 000C800018888200 action Add, label 19
3d17h: mpls_lbl_bind_chkpt: appl_id 13, KEY 000C800013200080
3d17h: mpls_chkpt_db: AVL insert successful, Key 000C800013200080 action Add, label 20
3d17h: mpls_lbl_bind_chkpt: appl_id 13, KEY 000C8000138383838200
3d17h: mpls_chkpt_db: AVL insert successful, Key 000C8000138383838200 action Add, label 21
3d17h: Stby RP OR CF peer not ready, don't send msg
3d17h: mpls_lbl_bind_chkpt: client ID 13 down, total client 0
3d17h: mpls_lbl_bind_chkpt: msg rx for 1D, vers 0, type 1
action 56, len 2, state 4, peer 13
3d17h: mpls_lbl_bind_chkpt: post msg type 1
3d17h: mpls_lbl_bind_chkpt: appl_id 13, KEY  action NSF unconfig, appl id 13
```

Related Commands

Command	Description
debug ip bgp vpnv4 checkpoint	Display the events for the VRF checkpointing system between the active and standby Route Processors.

debug mpls events

To display information about significant Multiprotocol Label Switching (MPLS) events, use the **debug mpls events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls events
no debug mpls events

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to monitor significant MPLS events.

Examples

The following is sample output from the **debug mpls events** command:

```
Router# debug mpls events

MPLS events debugging is on
TAGSW: Unbound IP address, 155.0.0.55, from Router ID
TAGSW: Bound IP address, 199.44.44.55, to Router ID
```

debug mpls infra label-broker api

To display Multiprotocol Label Switching (MPLS) label-broker API error messages, use the **debug mpls infra label-broker api** command in privileged EXEC mode. To disable the display of the messages, use the **no** form of this command.

debug mpls infra label-broker api [{**ipv4** | **ipv6** | [{**default** | **vrf** *vrf-name* }] | **prefix-list** { *prefix-name*}}]
no debug mpls infra label-broker api [{**ipv4** | **ipv6** | [{**default** | **vrf** *vrf-name* }] | **prefix-list** { *prefix-name*}}]

Syntax Description

ipv4	(Optional) Displays track labels for IPv4 prefixes.
ipv6	(Optional) Displays track labels for IPv6 prefixes.
default	(Optional) Displays the default routing/forwarding table.
vrf *vrf-name*	(Optional) Displays debugging information for the specified Virtual Private Network (VPN) routing and forwarding (VRF) instance. You can find VRF names using the **show ip vrf** command.
prefix-list	(Optional) Displays debugging information for the specified prefix list.
prefix-name	The name of the prefix list. You can find prefix list names using the **show ip prefix-list** command.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.8S	This command was introduced.

Usage Guidelines

To enable the **debug mpls infra label-broker api** command, the user must first enter global configuration mode and then enter the **service internal** command, followed by the **end** command.

Example

The following shows how to enable the **debug mpls infra label-broker api** command:

```
Device> enable
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# service internal
Device(config)# end
00:01:05: %SYS-5-CONFIG_I: Configured from console by console
Device# debug mpls infra label-broker api
MFI Label Broker:
  api debugging is on for all prefixes and labels
```

Related Commands

Command	Description
debug mpls infra label-broker api key	Displays information about the MFI label broker and track labels for key database entries.
debug mpls infra lfd label-block	Displays information about label-block debugging.
debug mpls infra lfd label-broker key	Displays information about keyed label debugging for all key entries.
service internal	Enables **infra** commands to be configured.
show ip prefix-list	Displays information about a prefix list or prefix list entries.
show ip vrf	Displays the set of defined VPN VRF instances and associated interfaces.
show xconnect	Displays information about xconnect attachment circuits and pseudowires.

debug mpls infra label-broker api key

To display Multiprotocol Label Switching (MPLS) application programming interface (API) key error messages, use the **debug mpls infra label-broker api key** command in privileged EXEC mode. To disable the display of the messages, use the **no** form of this command.

debug mpls infra label-broker api key [{ **vpnv4** | **vpnv6** | [{**rd** *ip-address*}]}] | [{ **per-vrf** [{**vrf** *vrf-name* | **default** | {**ipv4** | **ipv6**}}]}]
no debug mpls infra label-broker api key [{ **vpnv4** | **vpnv6** | [{**rd** *ip-address*}]}] | [{ **per-vrf** [{**vrf** *vrf-name* | **default** | {**ipv4** | **ipv6**}}]}]

Syntax Description

vpnv4	(Optional) Displays Virtual Private Network version 4 (VPNv4) events.
vpnv 6	(Optional) Displays Virtual Private Network version 6 (VPNv6) events.
rd	(Optional) Specifies a route distinguisher (RD) for a VPN routing and forwarding (VRF) instance.
ip-address	IPv4 or IPv6 address and mask.
per-vrf	(Optional) Specifies per-prefix label mode.
vrf *vrf-name*	(Optional) Displays debugging information for the specified Virtual Private Network (VPN) routing and forwarding (VRF) instance. You can find VRF names using the **show ip vrf** command.
ipv4	(Optional) Displays track labels for IPv4 prefixes.
ipv6	(Optional) Displays track labels for IPv6 prefixes.
default	(Optional) Displays the default routing/forwarding table.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.8S	This command was introduced.

Usage Guidelines

To enable the **debug mpls infra label-broker api key** command, the user must first enter global configuration mode, and then enter the **service internal** command, followed by the **end** command.

Example

The following shows how to enable the **debug mpls infra label-broker api key** command:

```
Device> enable
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# service internal
Device(config)# end
00:01:05: %SYS-5-CONFIG_I: Configured from console by console
Device# debug mpls infra label-broker api key
```

```
MFI Label Broker:
  api debugging is on for all IPv4 tables for IPv4 prefix list prefix-list
  api debugging is on for all IPv6 tables
  api debugging is on for all MPLS tables
  api debugging is on for all key entries
```

Related Commands

Command	Description
debug mpls infra label-broker api	Displays information about the MFI label broker and the API for all prefixes and labels.
debug mpls infra lfd label-block	Displays information about label-block debugging.
debug mpls infra lfd label-broker key	Displays information about keyed label debugging for all key entries.
service internal	Enables **infra** commands to be configured.
show ip prefix-list	Displays information about a prefix list or prefix list entries.
show ip vrf	Displays the set of defined VPN VRF instances and associated interfaces.
show xconnect	Displays information about xconnect attachment circuits and pseudowires.

debug mpls infra lfd label-block

To display information about label-block debugging, use the **debug mpls infra lfd label-block** command in privileged EXEC mode. To disable the display of the messages, use the **no** form of this command.

debug mpls infra lfd label-block [{**broker**}]
no debug mpls infra lfd label-block

Syntax Description

broker	(Optional) Displays debug messages for label-block broker events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.8S	This command was introduced.

Usage Guidelines

To enable the **debug mpls infra lfd label-block** command, the user must first enter global configuration mode, and then enter the **service internal** command, followed by the **end** command.

Example

The following shows how to enable the **debug mpls infra lfd label-block** command:

```
Device> enable
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# service internal
Device(config)# end
00:01:05: %SYS-5-CONFIG_I: Configured from console by console
Device# debug mpls infra lfd label-block
  label block debugging is on
```

Related Commands

Command	Description
debug mpls infra label-broker api	Displays information about the MFI label broker and the API for all prefixes and labels.
debug mpls infra label-broker api key	Displays information about the MFI label broker and track labels for key database entries.
debug mpls infra lfd label-broker key	Displays information about keyed label debugging for all key entries.
service internal	Enables **infra** commands to be configured.
show ip prefix-list	Displays information about a prefix list or prefix list entries

Command	Description
show ip vrf	Displays the set of defined VPN VRF instances and associated interfaces.
show xconnect	Displays information about xconnect attachment circuits and pseudowires.

debug mpls infra lfd label-broker key

To display information about keyed label debugging for all key entries, use the **debug mpls infra lfd label-broker key** command in privileged EXEC mode. To disable the display of the messages, use the **no** form of this command.

debug mpls infra lfd label-broker key [{**per-vrf** | [{**default** | **vrf** | {*vrf-name* | {**ipv4** | **ipv6**}}}] | | [{**vpnv4** | **vpnv6** | [{**rd** | {*ASN*:*nn* | *ip-address*:*nn* | {*ipv4-address-maskipv6-address-prefix*}}}]}]}]
no debug mpls infra lfd label-broker key [{**per-vrf** | [{**default** | **vrf** | {*vrf-name* | {**ipv4** | **ipv6**}}}] | | [{**vpnv4** | **vpnv6** | [{**rd** | {*ASN*:*nn* | *ip-address*:*nn* | {*ipv4-address-maskipv6-address-prefix*}}}]}]}]

Syntax Description

per-vrf	(Optional) Specifies per-prefix label mode.
default	(Optional) Displays the default routing/forwarding table.
vrf	Displays debugging information for the specified Virtual Private Network (VPN) routing and forwarding (VRF) instance.
vrf-name	The name of the VRF instance. You can find VRF names using the **show ip vrf** command.
ipv4	Displays track labels for IPv4 prefixes.
ipv6	Displays track labels for IPv6 prefixes.
vpnv4	(Optional) Displays Virtual Private Network version 4 (VPNv4) events.
vpnv6	(Optional) Displays Virtual Private Network version 6 (VPNv6) events.
rd	(Optional) Specifies a route distinguisher (RD) for a VRF instance.
asn:*nn*	IP address and network number.
ip-address:*nn*	Autonomous system number (ASN) and network number.
ipv4-address-mask	IPv4 address and subnet mask of the remote peer.
ipv6-address-prefix	IPv6 address and prefix length of the remote peer.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.8S	This command was introduced.

Usage Guidelines

To enable the **debug mpls infra lfd label-broker key** command, the user must first enter global configuration mode, and then enter the **service internal** command, followed by the **end** command.

Example

The following shows how to enable the **debug mpls infra lfd label-broker key** command:

```
Device> enable
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# service internal
Device(config)# end
00:01:05: %SYS-5-CONFIG_I: Configured from console by console
Device# debug mpls infra lfd label-broker key
  keyed label debugging is on for all key entries
```

Related Commands

Command	Description
debug mpls infra label-broker api	Displays information about the MFI label broker and the API for all prefixes and labels.
debug mpls infra label-broker api key	Displays information about the MFI label broker and track labels for key database entries.
debug mpls infra lfd label-block	Displays information about label-block debugging.
service internal	Enables **infra** commands to be configured.
show ip prefix-list	Displays information about a prefix list or prefix-list entries.
show ip vrf	Displays the set of defined VPN VRF instances and associated interfaces.
show xconnect	Displays information about xconnect attachment circuits and pseudowires.

debug mpls ip iprm

To display debugging information for the Multiprotocol Label Switching (MPLS) IP Rewrite Manager (IPRM), use the debug mpls ip iprm command in privileged EXEC mode. To disable the display of this information, use the **no** form of this command.

debug mpls ip iprm
no debug mpls ip iprm

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command displays all output related to IPRM.

Examples

The command in the following examples display all IPRM debugging for the global routing table.

Cisco 7500 Series Example

```
Router# debug mpls ip iprm
IPRM debugging is on for global routing table
  iprm: prefix deleted: 10.0.0.44/32(glbl)
  iprm: delete mfi rewrite: 10.0.0.44/32(glbl)
  .
  .
  .
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38002
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois
  iprm: announce prefix local labels: lcatm; trans #80; 10.0.0.44/32(glbl); 0 labels; flags
 0x0
  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm: omit rewrite create: 10.0.0.44/32(glbl)
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38000
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois
  iprm: announce prefix local labels: lcatm; trans #81; 10.0.0.44/32(glbl); 0 labels; flags
 0x0
```

```
  iprm: get path labels: 10.0.0.44/32(glbl); nh 10.0.0.55(glbl), Et4/0/1; trans #81; recurs
 tree change
  iprm: ldm get path labels: 10.0.0.44/32(glbl), ldp; flags 0x8000
  iprm: announce prefix local labels: ldp; trans #81; 10.0.0.44/32(glbl); 1 label; flags
0x0
  iprm:    lab 21, ltbl 0
  iprm: announce path labels: ldp; trans #81; 10.0.0.44/32(glbl); 0 labels; flags 0x0
  iprm:    path: nh 10.0.0.55(glbl), Et4/0/1
  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm:    lcl lab 21, ltbl 0, ldp
  iprm:    path lab -, nh 10.0.0.55(glbl), Et4/0/1; ldp
  iprm: create mfi rewrite 10.0.0.44/32(glbl) passed: 2 fpis/1 mois
  iprm:    fpi[0] IV4, owner IPRM; 10.0.0.44/32; glbl
  iprm:    fpi[1] LBL, owner LDP; 21, ltbl 0
  iprm:    moi[0] PKT, flags 0x8; lab label-no-label; nh 10.0.0.55; nh if Et4/0/1 (nsf)
```

Cisco 10000 Series Example

```
Router# debug mpls ip iprm
IPRM debugging is on for global routing table
  iprm: prefix deleted: 10.0.0.44/32(glbl)
  iprm: delete mfi rewrite: 10.0.0.44/32(glbl)
  .
  .
  .
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38002
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois

  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm: omit rewrite create: 10.0.0.44/32(glbl)
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38000
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois

  iprm: get path labels: 10.0.0.44/32(glbl); nh 10.0.0.55(glbl), GigabitEthernet4/0/0; trans
 #81; recurs tree change
  iprm: ldm get path labels: 10.0.0.44/32(glbl), ldp; flags 0x8000
  iprm: announce prefix local labels: ldp; trans #81; 10.0.0.44/32(glbl); 1 label; flags
0x0
  iprm:    lab 21, ltbl 0
  iprm: announce path labels: ldp; trans #81; 10.0.0.44/32(glbl); 0 labels; flags 0x0
  iprm:    path: nh 10.0.0.55(glbl), GigabitEthernet4/0/0

  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm:    lcl lab 21, ltbl 0, ldp
  iprm:    path lab -, nh 10.0.0.55(glbl), GigabitEthernet4/0/0; ldp
  iprm: create mfi rewrite 10.0.0.44/32(glbl) passed: 2 fpis/1 mois
  iprm:    fpi[0] IV4, owner IPRM; 10.0.0.44/32; glbl
  iprm:    fpi[1] LBL, owner LDP; 21, ltbl 0
  iprm:   moi[0] PKT, flags 0x8; lab label-no-label; nh 10.0.0.55; nh if GigabitEthernet4/0/0
 (nsf)
```

The table below describes the significant fields shown in the display. The field descriptions also apply to the output of following debug commands:

- **debug mpls ip iprm cef**
- **debug mpls ip iprm events**
- **debug mpls ip iprm ldm**
- **debug mpls ip iprm mfi**

Table 6: debug mpls ip iprm Field Descriptions

Field	Description
discover prefix labels	The prefix labels that the IP LDM discovered.
announce prefix local labels announce path labels	IP LDMs pass prefix incoming (local) and outgoing (path) labels to IPRM by announcing the labels.
mfi rewrite	The information required by MPLS Forwarding Infrastructure (MFI) to create forwarding data structures for an MPLS forwarding equivalence class (FEC). For IP over MPLS a prefix is an MPLS FEC. An MFI rewrite includes a set of forwarding path identifier (FPI) and MPLS output information (MOI) elements.
fpi	Forwarding path identifier, which is required to locate MPLS forwarding information for a FEC. IP over MPLS deals with several types of FPIs, including IPv4 (IV4), IPv6 (IV6), and label (LBL) FPIs. **Note** The Cisco 10000 series router does not support IPv6.
moi	MPLS output information. For IP over MPLS, there is a MOI for each prefix path. The MOI includes the next hop (nh), outgoing interface (nh if), and outgoing label. IP over MPLS handles several types of MOIs, including packet (PKT) and ATM (ATM).
get/create/update MFI rewrite	The process IPRM uses to read (get) or update (create/update) an MFI rewrite.
recurs tree change	Recursion tree change. Cisco Express Forwarding notifies IPRM when the recursion tree (see below) for a prefix changes. IPRM responds by performing label discovery (see above).
recursion tree	A prefix known to Cisco Express Forwarding can have one or more paths (routes). Each is either a terminal path with a next hop and an outgoing interface or a recursive path with a next hop and no outgoing interface. The next hop for a recursive path typically matches a prefix known to Cisco Express Forwarding. That prefix also has one or more paths. The IP recursion tree for prefix P is a tree rooted at P's Cisco Express Forwarding entry with one of more path descendants. Terminal paths are leaf nodes in P's recursion tree and recursive paths are nonleaf nodes, each of which points to the Cisco Express Forwarding entry for its next hop.
glbl	The global (default) routing table.
ctxt	Context. Information used by IPRM when it performs label discovery.
flags	Information passed between IPRM and other components.
trans #	Transaction number used to identify an ongoing label discovery.
ltbl	Label table.
nsf	Nonstop forwarding.

Related Commands

Command	Description
debug mpls ip iprm cef	Displays debugging information for interactions between Cisco Express Forwarding and the IPRM.
debug mpls ip iprm events	Displays events related to the MPLS IPRM.
debug mpls ip iprm ldm	Displays debugging information for interactions between the LDMs and the MPLS IPRM.
debug mpls ip iprm mfi	Displays debugging information for interactions between the MFI and the MPLS IPRM.

debug mpls ip iprm cef

To display debugging information for interactions between Cisco Express Forwarding and the Multiprotocol Label Switching (MPLS) IP Rewrite Manager (IPRM), use the debug mpls ip iprm cef command in privileged EXEC mode. To disable the display of these events, use the **no** form of this command.

debug mpls ip iprm cef [{**table** {**all***table-id*} | **vrf** *vrf-name* | **acl** *acl-name* | **prefix-list** *prefix-list-name*}]
no debug mpls ip iprm cef

Syntax Description

table	(Optional) Displays the debugging information for one or more routing tables.
all	Displays debugging information for all routing tables.
table-id	The ID of the routing table for which you want to display debugging information. Table 0 is the default or global routing table.
vrf	(Optional) Displays debugging information for the VPN routing and forwarding (VRF) instance you specify.
vrf-name	The name of the VRF instance. You can find VRF names with the show ip vrf command.
acl	(Optional) Displays debugging information for the access control list (ACL) you specify.
acl-name	The name of the ACL. You can find ACL names with the show ip access-list command.
prefix-list	(Optional) Displays debugging information for the prefix list you specify.
prefix-list-name	The name of the prefix list. You can find prefix list names with the show ip prefix-list command.

Command Default

Debugging is not enabled. If you do not supply an optional keyword, all the debugging events are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command limits the debug output to the IPRM interactions with Cisco Express Forwarding.

Examples

In the following example, IPRM events related to Cisco Express Forwarding are displayed.

Cisco 7500 Series Example

```
Router# debug mpls ip iprm cef
IPRM CEF interaction debugging is on for global routing table
iprm: prefix deleted: 10.0.0.44/32(glbl)
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38002
  iprm: announce prefix local labels: lcatm; trans #94; 10.0.0.44/32(glbl); 0 labels; flags
 0x0
  .
  .
  .
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38000
  iprm: announce prefix local labels: lcatm; trans #97; 10.0.0.44/32(glbl); 0 labels; flags
 0x0
  iprm: get path labels: 10.0.0.44/32(glbl); nh 10.0.0.55(glbl), Et4/0/1; trans #97; recurs
 tree change
  iprm: announce prefix local labels: ldp; trans #97; 10.0.0.44/32(glbl); 1 label; flags
0x0
  iprm:    lab 21, ltbl 0
  iprm: announce path labels: ldp; trans #97; 10.0.0.44/32(glbl); 0 labels; flags 0x0
  iprm:    path: nh 10.0.0.55(glbl), Et4/0/1
```

Cisco 10000 Series Example

```
Router# debug mpls ip iprm cef
IPRM CEF interaction debugging is on for global routing table
iprm: prefix deleted: 10.0.0.44/32(glbl)
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38002
  .
  .
  .
  iprm: discover prefix labels: 10.0.0.44/32(glbl); recurs tree change; ctxt 0x38000

  iprm: get path labels: 10.0.0.44/32(glbl); nh 10.0.0.55(glbl), GigabitEthernet4/0/0; trans
 #97; recurs tree change
  iprm: announce prefix local labels: ldp; trans #97; 10.0.0.44/32(glbl); 1 label; flags
0x0
  iprm:    lab 21, ltbl 0
  iprm: announce path labels: ldp; trans #97; 10.0.0.44/32(glbl); 0 labels; flags 0x0
  iprm:    path: nh 10.0.0.55(glbl), GigabitEthernet4/0/0
```

See the field descriptions for the **debug mpls ip iprm** command for an explanation of the fields displayed in the output.

Related Commands

Command	Description
debug mpls ip iprm events	Displays events related to the MPLS IPRM.
debug mpls ip iprm ldm	Displays debugging information for interactions between the IP LDMs and the MPLS IPRM.
debug mpls ip iprm mfi	Displays debugging information for interactions between the MFI and the MPLS IPRM.

debug mpls ip iprm events

To display events related to the Multiprotocol Label Switching (MPLS) IP Rewrite Manager (IPRM), use the debug mpls ip iprm events command in privileged EXEC mode. To disable the display of these events, use the **no** form of this command.

debug mpls ip iprm events
no debug mpls ip iprm events

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Examples

See the command page for **debug mpls ip iprm** for sample command output and an explanation of the fields displayed in the output.

Related Commands

Command	Description
debug mpls ip iprm cef	Displays debugging information for interactions between Cisco Express Forwarding and the IPRM.
debug mpls ip iprm ldm	Displays debugging information for interactions between the LDMs and the MPLS IPRM.
debug mpls ip iprm mfi	Displays debugging information for interactions between the MFI and the MPLS IPRM.

debug mpls ip iprm ldm

To display debugging information for interactions between the IP Label Distribution Modules (LDMs) and the Multiprotocol Label Switching (MPLS) IP Rewrite Manager (IPRM), use the debug mpls ip iprm ldm command in privileged EXEC mode. To disable the display of this information, use the **no** form of this command.

debug mpls ip iprm ldm [{**bgp** | **lcatm** | **ldp** | **vpnv4** | **6pe** | **table** {**all** | **table-id**} | **vrf vrf-name** | **acl acl-name** | **prefix-list prefix-list-name**}]
no debug mpls ip iprm ldm

Cisco 10000 Series Routers
debug mpls ip iprm ldm [{**bgp** | **ldp** | **vpnv4** | **table** {**all** | **table-id**} | **vrf vrf-name** | **acl acl-name** | **prefix-list prefix-list-name**}]
no debug mpls ip iprm ldm

Syntax Description

bgp	(Optional) Displays Border Gateway Protocol (BGP) events.
lcatm	(Optional) Displays Label Controlled ATM (LC-ATM) events. **Note** This keyword applies to Cisco 7000 series routers only.
ldp	(Optional) Displays Label Distribution Protocol (LDP) events.
vpnv4	(Optional) Displays Virtual Private Network (VPNv4) events.
6pe	(Optional) Displays IPv6 over MPLS events. **Note** This keyword applies to Cisco 7000 series routers only.
table	(Optional) Displays debugging information for one or more routing tables.
all	(Optional) Displays debugging information for all routing tables.
table-id	(Optional) Specifies the routing table for which you want to display debugging information. Table 0 is the default or global routing table.
vrf	(Optional) Displays debugging information for the VPN routing and forwarding (VRF) instance you specify.
vrf-name	(Optional) The name of the VRF instance. You can find VRF names with the show ip vrf command.
acl	(Optional) Displays debugging information for the access control list (ACL) you specify.
acl-name	(Optional) The name of the ACL. You can find ACL names with the show ip access-list command.
prefix-list	(Optional) Displays debugging information for the prefix list you specify.
prefix-list-name	(Optional) The name of the prefix list. You can find prefix list names with the show ip prefix-list command.

Command Default

Debugging is not enabled. If you do not supply an optional keyword, all the debugging events are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Examples

See the **debug mpls ip iprm** command page for sample output and an explanation of the fields displayed in the output.

Related Commands

Command	Description
debug mpls ip iprm cef	Displays debugging information for interactions between Cisco Express Forwarding and the IPRM.
debug mpls ip iprm events	Displays debugging information about events related to the MPLS IPRM.
debug mpls ip iprm mfi	Displays debugging information for interactions between the MFI and the MPLS IPRM.

debug mpls ip iprm mfi

To display debugging information for interactions between the Multiprotocol Label Switching (MPLS) Forwarding Infrastructure (MFI) and the MPLS IP Rewrite Manager (IPRM), use the debug mpls ip iprm mfi command in privileged EXEC mode. To disable the display of this information, use the **no** form of this command.

debug mpls ip iprm mfi [{**table** {**all***table-id*} | **vrf** *vrf-name* | **acl** *acl-name* | **prefix-list** *prefix-list-name*}]
no debug mpls ip iprm mfi

Syntax Description

table	(Optional) Displays debugging information for one or more routing tables.
all	(Optional) Displays debugging information for all routing tables.
table-id	(Optional) Displays debugging information for the routing table you specify. Table 0 is the default or global routing table.
vrf	(Optional) Displays debugging information for the VPN Routing and Forwarding (VRF) instance you specify.
vrf-name	(Optional) The name of the VRF instance. You can find VRF names with the show ip vrf command.
acl	(Optional) Displays debugging information for the access control list (ACL) you specify.
acl-name	(Optional) The name of the ACL. You can find ACL names with the show ip access-list command.
prefix-list	(Optional) Displays debugging information for the prefix list you specify.
prefix-list-name	(Optional) The name of the prefix list. You can find prefix list names with the show ip prefix-list command.

Command Default

Debugging is not enabled. If you enable debugging but do not supply an optional keyword, all the debugging events are displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Examples

The command in the following example displays MFI events.

Cisco 7500 Series Example

```
Router# debug mpls ip iprm mfi
IPRM MFI interaction debugging is on for global routing table
iprm: delete mfi rewrite: 10.0.0.44/32(glbl)
 .
 .
 .
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois
  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm: omit rewrite create: 10.0.0.44/32(glbl)
 .
 .
 .
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois
  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm:    lcl lab 21, ltbl 0, ldp
  iprm:    path lab -, nh 10.0.0.55(glbl), Et4/0/1; ldp
  iprm: create mfi rewrite 10.0.0.44/32(glbl) passed: 2 fpis/1 mois
  iprm:    fpi[0] IV4, owner IPRM; 10.0.0.44/32; glbl
  iprm:    fpi[1] LBL, owner LDP; 21, ltbl 0
  iprm:    moi[0] PKT, flags 0x8; lab label-no-label; nh 10.0.0.55; nh if Et4/0/1 (nsf)
```

Cisco 10000 Series Example

```
Router# debug mpls ip iprm mfi
IPRM MFI interaction debugging is on for global routing table
iprm: delete mfi rewrite: 10.0.0.44/32(glbl)
 .
 .
 .
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois
  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm: omit rewrite create: 10.0.0.44/32(glbl)
 .
 .
 .
  iprm: get mfi rewrite 10.0.0.44/32(glbl) obtained: 0 fpis/0 mois
  iprm: update mfi rewrite: 10.0.0.44/32(glbl); prefix label info
  iprm:    lcl lab 21, ltbl 0, ldp
  iprm:    path lab -, nh 10.0.0.55(glbl), GigabitEthernet4/0/0; ldp
  iprm: create mfi rewrite 10.0.0.44/32(glbl) passed: 2 fpis/1 mois
  iprm:    fpi[0] IV4, owner IPRM; 10.0.0.44/32; glbl
  iprm:    fpi[1] LBL, owner LDP; 21, ltbl 0
  iprm:    moi[0] PKT, flags 0x8; lab label-no-label; nh 10.0.0.55; nh if
GigabitEthernet4/0/0 (nsf)
```

See the **debug mpls ip iprm** command page for an explanation of the fields displayed in the output.

Related Commands

Command	Description
debug mpls ip iprm cef	Displays debugging information for interactions between Cisco Express Forwarding and the MPLS IPRM .

Command	Description
debug mpls ip iprm events	Displays events related to the MPLS IPRM.
debug mpls ip iprm ldm	Displays debugging information for interactions between the IP LDMs and the MPLS IPRM.

debug mpls l2transport checkpoint

To enable the display of Any Transport over MPLS (AToM) events when AToM is configured for nonstop forwarding/stateful switchover (NSF/SSO) and Graceful Restart, use the debug mpls l2transport checkpoint command in privileged EXEC mode. To disable the display of these messages, use the no form of this command.

debug mpls l2transport checkpoint
no debug mpls l2transport checkpoint

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging of the AToM NSF/SSO and Graceful Restart feature is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.

Usage Guidelines

Use debug commands with care. They use a significant amount of CPU time and can affect system performance.

Examples

In the following example, the output shows that NSF/SSO and Graceful Restart synchronize the data between the active and backup Route Processors after an AToM virtual circuit (VC) is created. (Both the **debug mpls l2transport checkpoint** and the **debug acircuit checkpoint** commands are enabled in this example.)

The debug mpls l2transport checkpoint command is enabled on the active RP:

```
Router# debug mpls l2transport checkpoint
Router# debug acircuit checkpoint
Router# show debug
AToM HA:
  AToM checkpointing events and errors debugging is on
AC HA:
  Attachment Circuit Checkpoint debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface Fa5/1/1.2
Router(config-subif)# xconnect 10.55.55.2 1002 pw-class mpls
AToM HA [10.55.55.2, 1002]: Build provision msg, SSM sw/seg 8192/8194 [0x2000/0x2002] PW
id 9216 [0x2400] local label 21
AC HA: Dynamic Sync. Event:4 Sw:8192[2000] Se:16385[4001]
AToM HA: CF sync send complete
AC HA CF: Sync send complete. Code:0
```

On the standby Route Processor, the following messages indicate that it receives checkpointing data:

```
AC HA [10.55.55.2, 1002]: Add to WaitQ. Flags:1
AToM HA [105.55.55.2, 1002]: Received 32-byte provision version 1 CF message
AC HA CF: ClientId:89, Entity:0 Length:40
AToM HA [10.55.55.2, 1002]: Process chkpt msg provision [1], ver 1
AToM HA [10.55.55.2, 1002]: Reserved SSM sw/seg 8192/8194 [0x2000/0x2002] PW id 9216 [0x2400]
AC HA: Process Msg:35586. Ptr:44CBFD90. Val:0
AC HA: Sync. Event:4 CktType:4 Sw:8192[2000] Se:16385[4001]
AC HA [10.55.55.2, 1002]: Remove from WaitQ. Flags:1[OK][OK]
```

During a switchover from the active to the backup Route Processor, the debug messages look similar to the following:

```
%HA-5-MODE: Operating mode is hsa, configured mode is sso.
AC HA RF: CId:83, Seq:710, Sta:RF_STATUS_OPER_REDUNDANCY_MODE_CHANGE, Opr:5, St:STANDBY
HOT, PSt:ACTIVE
AToM HA: CID 84, Seq 715, Status RF_STATUS_OPER_REDUNDANCY_MODE_CHANGE, Op 5, State STANDBY
 HOT, Peer ACTIVE
AC HA RF: CId:83, Seq:710, Sta:RF_STATUS_PEER_PRESENCE, Opr:0, St:STANDBY HOT, PSt:ACTIVE
AToM HA: CID 84, Seq 715, Status RF_STATUS_PEER_PRESENCE, Op 0, State STANDBY HOT, Peer
ACTIVE
AC HA RF: CId:83, Seq:710, Sta:RF_STATUS_PEER_COMM, Opr:0, St:STANDBY HOT, PSt:DISABLED
AToM HA: CID 84, Seq 715, Status RF_STATUS_PEER_COMM, Op 0, State STANDBY HOT, Peer DISABLED
%HA-2-CUTOVER_NOTICE: Cutover initiated. Cease all console activity until system restarts.
%HA-2-CUTOVER_NOTICE: Do not add/remove RSPs or line cards until switchover completes.
%HA-2-CUTOVER_NOTICE: Deinitializing subsystems...
%OIR-6-REMCARD: Card removed from slot 4, interfaces disabled
%OIR-6-REMCARD: Card removed from slot 5, interfaces disabled
%OIR-6-REMCARD: Card removed from slot 9, interfaces disabled
%HA-2-CUTOVER_NOTICE: Reinitializing subsystems...
%HA-2-CUTOVER_NOTICE: System preparing to restart...
%HA-5-NOTICE: Resuming initialization...
AC HA RF: CId:83, Seq:710, Sta:RF_STATUS_REDUNDANCY_MODE_CHANGE, Opr:7, St:STANDBY HOT,
PSt:DISABLED
.
.
.
%LDP-5-GR: LDP restarting gracefully.  Preserving forwarding state for 250 seconds.
AC HA RF: CId:83, Seq:710, Sta:RF_PROG_ACTIVE, Opr:0, St:ACTIVE, PSt:DISABLED
AToM HA: CID 84, Seq 715, Event RF_PROG_ACTIVE, Op 0, State ACTIVE, Peer DISABLED
AC HA: Process Msg:35588. Ptr:0. Val:0
AC HA: Switchover: Standby->Active
AC HA RF: Reconciling
```

Related Commands

Command	Description
debug acircuit checkpoint	Enables the display of AToM attachment circuit events when AToM is configured for NSF/SSO and Graceful Restart.

debug mpls l2transport fast-reroute

To enable the display of Fast Reroute debugging information, use the debug mpls l2transport fast-reroute command in privileged EXEC mode. To stop the display of these messages, use the no form of this command.

debug mpls l2transport fast-reroute commanddebug mpls l2transport fast-reroute
no debug mpls l2transport fast-reroute

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging of the fast reroute feature is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(25)S	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command does not display output on platforms where AToM Fast Reroute is implemented in the forwarding code. This command does not display output for the Cisco 7500 (both RP and VIP) series routers, 7200 series routers, and Cisco 12000 series route processor. The command does display output on Cisco 10720 Internet router line cards and Cisco 12000 series line cards.

Examples

In the following example, the primary link is disabled, which causes the backup tunnel (Tu1) to become the primary path.

```
Router# execute-on slot 3 debug mpls l2transport fast-reroute
========= Line Card (Slot 3) =========
AToM fast reroute debugging is on
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Processing TFIB FRR event for 1.4.0.1
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Finished processing TFIB FRR event for 1.4.0.1
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Processing TFIB FRR event for Tunnel41
SLOT 3:Sep 16 17:58:56.346: AToM SMGR: Finished processing TFIB FRR event for Tunnel41
Sep 16 17:58:58.342: %LINK-3-UPDOWN: Interface POS0/0, changed state to down
Sep 16 17:58:58.342: %OSPF-5-ADJCHG: Process 1, Nbr 1.0.0.1 on POS0/0 from FULL to DOWN,
Neighbor Down: Interface down or detached
Sep 16 17:58:59.342: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS0/0, changed state
 to down
```

Related Commands

Command	Description
show mpls traffic-eng fast-reroute database	Displays the contents of the Fast Reroute database.

debug mpls l2transport ipc

To display the interprocessor communication (IPC) messages exchanged between distributed platforms, such as the Cisco 12000 series router and the Cisco 7500 series routers, use the **debug mpls l2transport ipc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport ipc
no debug mpls l2transport ipc

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

You can issue this command either from the line card or the route processor to log Any Transport over MPLS (AToM) updates to or from line cards. This command applies only to platforms that support distributed mode.

Examples

The following is sample output from the **debug mpls l2transport ipc** command:

```
Router# debug mpls l2transport ipc
AToM ipc debugging is on
*May 27 23:56:04.699 UTC: AToM SMGR: Repopulating line card 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Sending Imposition update to slot
 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Imposition being done on ingress
interface
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Sending disposition update to slot
 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1101]: Distributing disposition info to
all linecards
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Sending Imposition update to slot
255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Imposition being done on ingress
interface
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Sending disposition update to slot
 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 701]: Distributing disposition info to
all linecards
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Sending Imposition update to slot
 255
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Imposition being done on ingress
interface
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Sending disposition update to slot
 255
```

```
*May 27 23:56:04.699 UTC: AToM SMGR [17.17.17.17, 1201]: Distributing disposition info to
all linecards
```

debug mpls l2transport packet

To display information about the status of Any Transport over MPLS (AToM) switched packets, use the **debug mpls l2transport packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport packet {**data** | **error**}
no debug mpls l2transport packet {**data** | **error**}

Syntax Description

data	Displays (in hex) the AToM switched packets for imposition and disposition. This can help validate that packets are flowing between the customer edge (CE) routers. Also, you can display the packets to check the format of the data or the data itself.
error	Displays AToM switching errors, such as the reason that packets cannot be switched. This can help identify why data is not being transported.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command sparingly because the command output can be overwhelming.

For platforms that support distributed switching, the command displays output only for packets switched by the central route processor module. Packets switched autonomously by the linecards are not displayed. For example, packets switched by Versatile Interface Processors (VIPs) on the Cisco 7500 router are not displayed.

Examples

The following is sample output from the **debug mpls l2transport packet** commands for a PPP over MPLS configuration:

```
Router# debug mpls l2transport packet data
AToM packet data debugging is on
Router# debug mpls l2transport packet error
AToM packet errors debugging is on
Router# show debug

AToM:
  AToM packet data debugging is on
  AToM packet errors debugging is on
*Mar 24 23:29:30.495: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:30.495: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:30.495: 0F 00 88 47 00 01 10 FF 00 01 51 02 00 00 00 00
*Mar 24 23:29:30.495: 00 FD C0 01 01 01 C0 4B 41 73 F4 00 01 00 02 CC
*Mar 24 23:29:30.495: 66 51 88 B4 CE 73 39 00 00 40 00 88 03 02 00 70
```

```
*Mar 24 23:29:30.495: 23 30 00 04 3C 61 83 C0 00 06 00 06 94 CC A7 23
*Mar 24 23:29:30.495: 49 84 D8 33 17 8C F2 60 00 11 9E 80 00 50 08 08
*Mar 24 23:29:30.495: 86 69 39 98 CD E2 02 49 B8 E9 9D 0D C6 53 A1 DC
*Mar 24 23:29:30.495: DE 72 35 88 09 E7 0C 60 61 3A 1A 4D C6 71 01 4C
*Mar 24 23:29:30.495: F2 73 CC 06 DC 38 6F 33 66 83 09 C8 CA 20 05 12
*Mar 24 23:29:30.495: 49 E5 31 00 A0 E8 6D 14 88 06 E3 21 80 C3 31 E4
*Mar 24 23:29:30.495: 28 21 E4 21 69 28 A6 2D 26 8A 45 82 02 B6 FC 39
*Mar 24 23:29:30.499: D8 60 A3 62 B1 60 A5 80
*Mar 24 23:29:31.835: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:31.835: FF 03 00 FD C0 04 8A 57 FF FF FF FF FF FF FF FF
*Mar 24 23:29:31.835: FF FF FB 14 B0 00
*Mar 24 23:29:49.423: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.423: FF 03 C0 21 01 11 00 0F 03 05 C2 23 05 05 06 5F
*Mar 24 23:29:49.423: 23 35 D4
*Mar 24 23:29:49.435: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.435: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.435: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 15 00 00
*Mar 24 23:29:49.435: C0 21 01 2F 00 0F 03 05 C2 23 05 05 06 5F CC 5F
*Mar 24 23:29:49.435: E5
*Mar 24 23:29:49.435: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.435: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.435: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 15 00 00
*Mar 24 23:29:49.435: C0 21 02 11 00 0F 03 05 C2 23 05 05 06 5F 23 35
*Mar 24 23:29:49.435: D4
*Mar 24 23:29:49.443: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.443: FF 03 C0 21 02 2F 00 0F 03 05 C2 23 05 05 06 5F
*Mar 24 23:29:49.443: CC 5F E5
*Mar 24 23:29:49.447: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.447: FF 03 C2 23 01 D0 00 1C 10 45 59 13 1A 92 FD 93
*Mar 24 23:29:49.447: 01 A2 CF B6 FB 3A 04 46 93 63 65 32 2D 67 73 72
*Mar 24 23:29:49.451: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.451: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.451: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 22 00 00
*Mar 24 23:29:49.451: C2 23 01 F5 00 1C 10 F1 98 35 3F 79 F2 1A 15 10
*Mar 24 23:29:49.451: B4 C0 73 D7 B1 9F 2A 63 65 31 2D 67 73 72
*Mar 24 23:29:49.455: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.455: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.455: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 22 00 00
*Mar 24 23:29:49.455: C2 23 02 D0 00 1C 10 56 4A 32 5B 99 55 D5 CF 44
*Mar 24 23:29:49.455: FC D3 D9 3F CC 8C A8 63 65 31 2D 67 73 72
*Mar 24 23:29:49.463: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.463: FF 03 C2 23 02 F5 00 1C 10 45 84 E4 E5 DD C0 5F
*Mar 24 23:29:49.463: FD 2F 37 63 9A 3D 03 7B B9 63 65 32 2D 67 73 72
*Mar 24 23:29:49.463: ATOM-L2 Switching Disposition Packet data:
*Mar 24 23:29:49.463: FF 03 C2 23 03 D0 00 04
*Mar 24 23:29:49.471: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.471: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.471: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 0A 00 00
*Mar 24 23:29:49.471: C2 23 03 F5 00 04
*Mar 24 23:29:49.471: ATOM-PPP Switching: check features failed.
*Mar 24 23:29:49.471: ATOM-PPP Switching (Fast) Imposition Packet data: experimental bits
are 0
*Mar 24 23:29:49.471: 0F 00 88 47 00 01 10 FF 00 01 61 02 00 10 00 00
*Mar 24 23:29:49.471: 80 21 01 0B 00 0A 03 06 78 01 01 78
*Mar 24 23:29:49.475: ATOM-PPP Switching: check features failed.
```

debug mpls l2transport signaling

To display information about the Any Transport over MPLS (AToM) signaling protocol, use the **debug mpls l2transport signaling** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport signaling {**event** | **message**}
no debug mpls l2transport signaling {**event** | **message**}

Syntax Description

event	Displays AToM signaling events.
message	Displays AToM signaling status messages.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug mpls l2transport signaling** command:

```
Router# debug mpls l2transport signaling event
AToM LDP event debugging is on
Router# debug mpls l2transport signaling message
AToM LDP message debugging is on
Router# show debugging
AToM:
  AToM LDP event debugging is on
  AToM LDP message debugging is on
*Mar 24 23:10:55.611: AToM LDP [9.9.9.9]: Allocate LDP instance
*Mar 24 23:10:55.611: AToM LDP [9.9.9.9]: Opening session, 1 clients
*Mar 24 23:10:56.063: %SYS-5-CONFIG_I: Configured from console by console
*Mar 24 23:10:56.583: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed
state to up
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Session is up
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 1.1.1.100
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 46.1.1.6
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 9.9.9.9
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Peer address change, add 57.1.1.6
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Sending label mapping msg
vc type 7, cbit 1, vc id 50, group id 6, vc label 21, status 0, mtu 1500
*Mar 24 23:11:00.539: AToM LDP [9.9.9.9]: Received label mapping msg, id 113
vc type 7, cbit 1, vc id 50, group id 6, vc label 21, status 0, mtu 1500
```

debug mpls l2transport static-oam

To enable the display of messages related to static pseudowire operations administrative and management (OAM), use the **debug mpls l2transport static-oam** command in privileged EXEC mode. To disable the display of these messages, use the **no** form of this command.

debug mpls l2transport static-oam [{**elog** | **error** | **event** | **fsm**}]
no debug mpls l2transport static-oam

Syntax Description

elog	Displays logging messages for static pseudowire OAM.
error	Displays error messages for static pseudowire OAM.
event	Displays event messages for static pseudowire OAM.
fsm	Displays finite state machine (FSM) messages for static pseudowire OAM.

Command Default

Static sseudowire messages are not displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(1)SA	This command was introduced.
15.1(3)S	This command was integrated.

Examples

The following example enables the display of error messages for static pseudowire OAM:

```
Router# debug mpls l2transport static-oam error
```

Related Commands

Command	Description
show mpls l2transport static-oam	Displays the status of static pseudowires.

debug mpls l2transport vc

To display information about the status of the Any Transport over MPLS (AToM) virtual circuits (VCs), use the**debug mpls l2transport vc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport vc {**event** | **fsm** | **ldp** | **sss** | **status** {**event** | **fsm**}}
no debug mpls l2transport vc {**event** | **fsm** | **ldp** | **sss** | **status** {**event** | **fsm**}}

Syntax Description

event	Displays AToM event messages about the VCs.
fsm	Displays debug information related to the finite state machine (FSM).
ldp	Displays debug information related to the Label Distribution Protocol (LDP).
sss	Displays debug information related to the subscriber service switch (SSS).
status	Displays debug information related to the status of the VCs.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.0(25)S	This command was integrated into Cisco IOS Release 12.0(25)S.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SRC	The command was updated to include the **ldp**, **sss**, and **status** keywords as part of the MPLS Pseudowire Status Signaling feature.
12.2(50)SY	This command was integrated into Cisco IOS Release 12.2(50)SY.

Usage Guidelines

You can issue this command from the line card or the route processor.

Examples

The following is sample output from the **debug mpls l2transport vc** command:

```
Router# debug mpls l2transport vc event
```

```
AToM vc event debugging is on
Router# debug mpls l2transport vc fsm
AToM vc fsm debugging is on
Router# show debugging

AToM:
  AToM vc event debugging is on
  AToM vc fsm debugging is on
*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Event provision, state changed from idle to
 provisioned
*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Provision vc
*Mar 24 23:17:24.371: AToM SMGR [10.9.9.9, 50]: Requesting VC create, vc_handle 61A09930
*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Event local up, state changed from provisioned
 to local standby
*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Update local vc label binding
*Mar 24 23:17:24.371: AToM SMGR [10.9.9.9, 50]: sucessfully processed create request
*Mar 24 23:17:24.875: %SYS-5-CONFIG_I: Configured from console by console
*Mar 24 23:17:25.131: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed
state to up
*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Event ldp up, state changed from local standby
 to local ready
*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Advertise local vc label binding
*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Event remote up, state changed from local
ready to establishing
*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Remote end up
*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Event remote validated, state changed from
establishing to established
*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Validate vc, activating data plane
*Mar 24 23:17:28.567: AToM SMGR [10.9.9.9, 50]: Processing imposition update, vc_handle
61A09930, update_action 3, remote_vc_label 21
*Mar 24 23:17:28.567: AToM SMGR [10.9.9.9, 50]: Imposition Programmed, Output Interface:
PO5/0
*Mar 24 23:17:28.567: AToM SMGR [10.9.9.9, 50]: Processing disposition update, vc_handle
61A09930, update_action 3, local_vc_label 22
*Mar 24 23:17:28.571: AToM SMGR: Processing TFIB event for 10.9.9.9
*Mar 24 23:17:28.571: AToM SMGR [10.9.9.9, 50]: Imposition Programmed, Output Interface:
PO5/0
```

The following is sample output of MPLS Pseudowire Status Signaling messages from the **debug mpls l2transport vc status event and debug mpls l2transport vc status fsm**commands:

```
Router# debug mpls l2transport vc status event
Router# debug mpls l2transport vc status fsm
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: Receive SSS STATUS(UP)
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: AC status UP
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Evt local up, LndRru->LnuRru
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Evt local ready, LnuRru->LruRru
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Act send label(UP)
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: Send label(UP)
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: Local AC : UP
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: Dataplane: no fault
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: Overall  : no fault
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: Remote label is ready
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Evt remote ready in LruRru
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Evt remote up in LruRru
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Evt dataplane clear fault in LruRru
*Feb 26 14:03:42.543: AToM MGR [10.9.9.9, 100]: S:Evt dataplane clear fault in LruRru
*Feb 26 14:03:42.551: AToM MGR [10.9.9.9, 100]: S:Evt dataplane clear fault in LruRru
```

The status codes in the messages, such as S: and LruRru, indicate the staus of the local and remote routers. The following list translates the status codes:

L--local router

R--remote router

r or n--ready (r) or not ready (n)

u or d-- up (u) or down (d) status

The output also includes the following values:

D--Dataplane

S--Local shutdown

debug mpls l2transport vc subscriber

To enable debugging for Any Transport over MPLS (AToM) virtual circuit (VC) subscriber sessions, use the **debug mpls l2transport vc subscriber**command in privileged EXEC mode. To disable debugging for AToM VC subscribers, use the **no** form of this command.

debug l2transport vc subscriber {**error** | **event**}

Syntax Description

error	Specifies debugging for AToM VC subscriber session errors.
event	Specifies debugging for AToM VC subscriber session events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(2)S	This command was introduced.

Usage Guidelines

The **debug l2transport vc subscriber** command displays the debugs for flows involving the creation of AToM VCs as a result of the detection of First Sign of Life (FSOL) events.

Examples

The following is sample output from the **debug l2transport vc subscriber** command:

```
Router# debug mpls l2transport vc subscriber error
AToM LDP subscriber error debugging is on
Router# debug mpls l2transport vc subscriber event
AToM LDP subscriber event debugging is on
Router# show debugging
AToM:
  AToM vc event debugging is on
  AToM LDP subscriber event debugging is on
  AToM LDP subscriber error debugging is on
Router# show logging
Syslog logging: enabled (0 messages dropped, 41 messages rate-limited, 0 flushes, 0 overruns,
 xml disabled, filtering disabled)
No Active Message Discriminator.
No Inactive Message Discriminator.
    Console logging: level debugging, 498 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 229 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 123 message lines logged
Log Buffer (10000000 bytes):
*Apr 15 20:58:34.048: AToM LDP: Receive label adv
*Apr 15 20:58:34.048: AToM[33.33.33.34, 12346]: Received LDP label msg, signal peer ID
0.0.0.0
*Apr 15 20:58:34.048: AToM[33.33.33.34, 12346]: label details: type 5, label 20
*Apr 15 20:58:34.048: AToM LDP Sub::Found subscriber author group atom_test1, for peer ID
33.33.33.34, VC ID 12346
```

```
*Apr 15 20:58:34.048: AToM LDP Sub::33.33.33.34 created pre-auth key
peer-ip:33.33.33.34:vc-id:12346
*Apr 15 20:58:34.049: AToM LDP Sub::SDB find string/context not found
31:peer-ip:33.33.33.34:vc-id:12346
*Apr 15 20:58:34.049: AToM LDP Sub::creating FSOL context - string
31:peer-ip:33.33.33.34:vc-id:12346, service 0
*Apr 15 20:58:34.049: AToM LDP Sub::SDB add string success 31:peer-ip:33.33.33.34:vc-id:12346,
 0x43EFE12
*Apr 15 20:58:34.049: AToM LDP Sub::Init notify 31:peer-ip:33.33.33.34:vc-id:12346
*Apr 15 20:58:34.049: AToM LDP Sub::SDB get FSOL handle success 0x87E7000, 0x43EFE12
*Apr 15 20:58:34.049: AToM LDP Sub::SDB sanity check success 0x87E7000, 0x83928940
*Apr 15 20:58:34.049: AToM LDP Sub::SDB found string/context
31:peer-ip:33.33.33.34:vc-id:12346, 0x43EFE12
*Apr 15 20:58:34.049: AToM LDP Sub::find/create fsol found fsol after add :
*Apr 15 20:58:34.049: AToM LDP Sub::71237138(0x43EFE12) 31:peer-ip:33.33.33.34:vc-id:12346
*Apr 15 20:58:34.049: AToM LDP Sub::find/create fsol context success
*Apr 15 20:58:34.050: AToM LDP Sub::Preauth request success, handle 0x43EFE12, AAA ID 0x14,
 policy handle 0x5100000C
*Apr 15 20:58:34.050: AToM[33.33.33.34, 12346]: Succeeded to make pre-author request for
ATOM LDP FSOL
*Apr 15 20:58:34.050: AToM[33.33.33.34]: status notification failed: no such vc
*Apr 15 20:58:34.075: AToM LDP Sub::handle 0x43EFE12, AAA attribute 1054
*Apr 15 20:58:34.075: AToM LDP Sub::AAA attribute 1080 not handled
*Apr 15 20:58:34.075: AToM LDP Sub::AAA attribute 968 not handled
*Apr 15 20:58:34.075: AToM LDP Sub::handle 0x43EFE12, AAA attribute 1075
*Apr 15 20:58:34.075: AToM LDP Sub::handle 0x43EFE12, AAA attribute 1056
*Apr 15 20:58:34.075: AToM LDP Sub::handle 0x43EFE12, attribute 1056 val 1
*Apr 15 20:58:34.075: AToM LDP Sub::handle 0x43EFE12, protocol 1
*Apr 15 20:58:34.075: AToM LDP Sub::added 0x8152A00 to 0x370000C7 successfully
*Apr 15 20:58:34.075: AToM LDP Sub::handle 0x43EFE12, Pre-author parser returning 0
*Apr 15 20:58:34.098: AToM LDP Sub::handle 0x43EFE12, AAA attribute 1054
*Apr 15 20:58:34.098: AToM LDP Sub::AAA attribute 1080 not handled
*Apr 15 20:58:34.099: AToM LDP Sub::AAA attribute 968 not handled
*Apr 15 20:58:34.099: AToM LDP Sub::handle 0x43EFE12, AAA attribute 1075
*Apr 15 20:58:34.099: AToM LDP Sub::handle 0x43EFE12, AAA attribute 1056
*Apr 15 20:58:34.099: AToM LDP Sub::handle 0x43EFE12, attribute 1056 val 1
*Apr 15 20:58:34.099: AToM LDP Sub::handle 0x43EFE12, protocol 1
*Apr 15 20:58:34.099: AToM LDP Sub::added 0x81529A0 to 0xB50000D0 successfully
*Apr 15 20:58:34.099: AToM LDP Sub::handle 0x43EFE12, Pre-author parser returning 0
*Apr 15 20:58:34.099: AToM LDP Sub::Preauth callback for client 0x43EFE12, AAA 0x14
*Apr 15 20:58:34.100: AToM LDP Sub::SDB get FSOL handle success 0x87E7000, 0x43EFE12
*Apr 15 20:58:34.100: AToM LDP Sub::SDB sanity check success 0x87E7000, 0x83928940
*Apr 15 20:58:34.100: AToM LDP Sub::atom preauth callback 0. processing info 108
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14, processing info type
108, val 0x81529A0, list 0xB50000D0
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14, attribute 1054 12346 len
 5, VC ID 12346
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14 attribute 1075 len 555819298
 33.33.33.34
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14, attribute 1056 val 1
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14, method 3, protocol 4
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14 connect notifyfor VPWS
service
*Apr 15 20:58:34.100: AToM LDP Sub::handle 0x43EFE12, AAA ID 0x14, added member to provision
 VPWS service
*Apr 15 20:58:34.100: AToM LDP Sub::downloaded attribute parsing success, handle 0x43EFE12,
 AAA ID 0x14
*Apr 15 20:58:34.124: AToM LDP Sub::
atom_ldp_subscriber_parse_preauth1char = p:31
*Apr 15 20:58:34.124: AToM LDP Sub::
atom_ldp_subscriber_parse_preauth2char = 3:23
*Apr 15 20:58:34.124: AToM LDP Sub::
atom_ldp_subscriber_parse_preauth3char = v:11:11
*Apr 15 20:58:34.124: AToM LDP Sub::
```

```
atom_ldp_subscriber_parse_preauth4char = 33.33.33.34
*Apr 15 20:58:34.124: AToM LDP Sub::0.0.0.0: parsed 33.33.33.34
*Apr 15 20:58:34.124: AToM LDP Sub::
atom_ldp_subscriber_parse_preauth5char = v:11
*Apr 15 20:58:34.124: AToM LDP Sub::
atom_ldp_subscriber_parse_preauth6char = 1:5
*Apr 15 20:58:34.124: AToM LDP Sub::
atom_ldp_subscriber_parse_preauth7char = 12346
*Apr 15 20:58:34.124: AToM LDP Sub::33.33.33.34: parsed VC ID 12346
*Apr 15 20:58:34.124: AToM LDP Sub::Found subscriber author group atom_test1, for peer ID
33.33.33.34, VC ID 12346
*Apr 15 20:58:34.124: AToM LDP Sub::SDB get FSOL handle success 0x87E7000, 0x43EFE12
*Apr 15 20:58:34.124: AToM LDP Sub::SDB sanity check success 0x87E7000, 0x83928940
*Apr 15 20:58:34.124: AToM LDP Sub::SDB found string/context
31:peer-ip:33.33.33.34:vc-id:12346, 0x43EFE12
*Apr 15 20:58:34.125: AToM LDP Sub::found context from earlier trigger, 1, 0, ignoring this
 request
*Apr 15 20:58:34.125: AToM LDP Sub::Found existing FSOL context ID
31:peer-ip:33.33.33.34:vc-id:12346, re-use 1, 0
*Apr 15 20:58:34.156: AToM[33.33.33.34, 12346]: Provisioned
*Apr 15 20:58:34.156: AToM[33.33.33.34, 12346]: Evt provision, idle -> provisioned
*Apr 15 20:58:34.156: AToM[33.33.33.34, 12346]: . Provision vc
*Apr 15 20:58:34.156: AToM LDP[33.33.33.34, 12346]: LDP OPEN request
*Apr 15 20:58:34.156: AToM LDP[33.33.33.34, 12346]: Signaling peer-id of VC changed to
33.33.33.34
*Apr 15 20:58:34.156: AToM[33.33.33.34, 12346]: Evt remote ready, provisioned -> remote
ready
*Apr 15 20:58:34.156: AToM[33.33.33.34, 12346]: . Receive remote vc label binding, instance
 3
*Apr 15 20:58:34.160: AToM[33.33.33.34, 12346]: Receive SSS CONNECT
*Apr 15 20:58:34.160: AToM[33.33.33.34, 12346]: . Update AIE peer 9F00000F our 43000010
*Apr 15 20:58:34.161: AToM[33.33.33.34, 12346]: ... Evt local ready, remote ready ->
establishing
*Apr 15 20:58:34.161: AToM[33.33.33.34, 12346]: ..... Alloc local binding
*Apr 15 20:58:34.161: AToM[33.33.33.34, 12346]: ..... autosense disabled [init]
*Apr 15 20:58:34.161: AToM[33.33.33.34, 12346]: ..... autosense enabled
*Apr 15 20:58:34.161: AToM[33.33.33.34, 12346]: ..... Grouping on (value 1)
*Apr 15 20:58:34.161: AToM[33.33.33.34, 12346]: ..... Grouping ignored, set to 0
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: ..... MTU set to 1500
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: .... Local end available
*Apr 15 20:58:34.162: AToM LDP[33.33.33.34, 12346]: Send label(DOWN)
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: .... Validate remote binding
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: ..... Evt remote validated in establishing
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: ..... Validate vc, activating data plane
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: ..... Update peer with our circuit type
Eth
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: ..... Check if can activate dataplane
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: .. Check if can activate dataplane
*Apr 15 20:58:34.162: AToM[33.33.33.34, 12346]: . Send SSS CONNECTED
*Apr 15 20:58:34.163: AToM[33.33.33.34, 12346]: ... No PW Status SP info to report to SSS
peer
*Apr 15 20:58:34.163: AToM LDP[33.33.33.34, 12346]: LDP open
*Apr 15 20:58:34.163: AToM LDP[33.33.33.34, 12346]: Signaling peer-id of VC changed to
33.33.33.34
*Apr 15 20:58:34.163: AToM LDP[33.33.33.34, 12346]: LDP UP
*Apr 15 20:58:34.163: AToM[33.33.33.34, 12346]: Evt ldp up in establishing
*Apr 15 20:58:34.163: AToM[33.33.33.34, 12346]: . Take no action
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: Receive SSS FSP STATUS
*Apr 15 20:58:34.169: AToM LDP[33.33.33.34, 12346]: Send notify(UP)
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: ... Evt local ready in establishing
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: .... Take no action
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: .. Check if can activate dataplane
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: ... Attempt to activate dataplane on Active
 RP, VC in establishing state
```

```
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: ... Evt dataplane activate, establishing
-> activating
*Apr 15 20:58:34.169: AToM[33.33.33.34, 12346]: .... Activating data plane
*Apr 15 20:58:34.170: AToM[33.33.33.34, 12346]: .... Activate dataplane
*Apr 15 20:58:34.170: AToM[33.33.33.34, 12346]: .... Need to setup the dataplane
*Apr 15 20:58:34.170: AToM[33.33.33.34, 12346]: .... Setup dataplane
*Apr 15 20:58:34.170: AToM[33.33.33.34, 12346]: ..... Same peer; get switch hdl 4100
*Apr 15 20:58:34.170: AToM[33.33.33.34, 12346]: ..... Set segment count to 1
*Apr 15 20:58:34.170: AToM[33.33.33.34, 12346]: ..... Provision SSM with 4100/8203 (sw/seg)
*Apr 15 20:58:34.173: AToM[33.33.33.34, 12346]: Receive SSM dataplane up notification
*Apr 15 20:58:34.174: AToM[33.33.33.34, 12346]: Receive SSM dataplane up notification
*Apr 15 20:58:34.174: AToM[33.33.33.34, 12346]: Evt dataplane up, activating -> established
*Apr 15 20:58:34.174: AToM[33.33.33.34, 12346]: . Dataplane activated
*Apr 15 20:58:34.174: AToM[33.33.33.34, 12346]: SYSLOG: VC is UP
*Apr 15 20:58:34.174: AToM[33.33.33.34, 12346]: Evt dataplane up in established
*Apr 15 20:58:34.174: AToM[33.33.33.34, 12346]: . Take no action
*Apr 15 20:58:42.222: AToM[33.33.33.34, 12346]: Label 23 freed
```

Related Commands

Command	Description
show debugging	Displays information about the types of debugging that are enabled for your router.
show logging	Displays the state of system logging (syslog) and the contents of the standard system logging buffer.
show mpls l2transport vc	Displays information about the status of the AToM VCs.

debug mpls l2transport vc vccv

To enable Any Transport over MPLS (AToM) Virtual Circuit Connection Verification (VCCV) debugging, use the **debug mpls l2transport vc vccv**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls l2transport vc vccv [**bfd**] **event**
no debug mpls l2transport vc vccv [**bfd**] **event**

Syntax Description

bfd	(Optional) Displays event messages when Bidirectional Forwarding Detection (BFD) sessions are created, when BFD sends dataplane fault notifications to Layer 2 VPN (L2VPN), and when L2VPN sends the attachment circuit (AC) signaling status to BFD.
event	Displays AToM event messages about the VCCV.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command to enable AToM VCCV events and AToM VCCV BFD events debugging.

Examples

The following examples show how to enable MPLS L2transport VC VCCV and VCCV BFD event debugging:

```
Router# debug mpls l2transport vc vccv bfd event
AToM VCCV BFD events debugging is on
Router# debug mpls l2transport vc vccv event
AToM VCCV events debugging is on
Router# show debugging
AToM VCCV BFD events debugging is on
AToM VCCV events debugging is on
```

Related Commands

Command	Description
show mpls l2transport vc	Displays information about the status of the AToM VCs.

debug mpls ldp advertisements

To display information about the advertisement of labels and interface addresses to label distribution protocol (LDP) peers, use the **debugmplsldpadvertisements**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls ldp advertisements [**peer-acl** *acl*] [**prefix-acl** *acl*]
no debug mpls ldp advertisements [**peer-acl** *acl*] [**prefix-acl** *acl*]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed advertisements to those for LDP peers permitted by the access control list (*acl*).
prefix-acl *acl*	(Optional) Limits the displayed advertisements to those for prefixes permitted by the access control list (*acl*).

Command Default

Displays information about advertisements to all LDP peers for all prefixes.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to monitor the label and address advertisements to LDP peers.

Use the **peer-acl** or **prefix-acl** options separately or together to limit the information display to specific LDP peers and/or specific prefixes.

Note This command monitors advertisement of non-LC-ATM labels (generic labels) only. Use the **debugmplsatm-ldp** command to monitor LC-ATM activity.

Examples

The following shows sample output from the **debugmplsldpadvertisements** command:

```
Router# debug mpls ldp advertisements
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 130.77.0.33
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 133.0.0.33
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 34.0.0.33
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 103.0.0.33
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 35.0.0.33
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 38.0.0.33
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 34.0.0.0/8, label 3 (#2)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 203.0.7.7/32, label 24 (#4)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 35.0.0.0/8, label 3 (#8)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 103.0.0.0/8, label 3 (#10)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 138.1.0.0/16, label 26 (#14)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 155.0.0.55/32, label 27 (#16)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 38.0.0.0/8, label 3 (#18)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 212.10.1.0/24, label 30 (#24)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 59.0.0.0/8, label 32 (#28)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 144.0.0.44/32, label 33 (#30)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 106.0.0.0/8, label 34 (#32)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 133.0.0.33/32, label 3 (#34)
tagcon: peer 144.0.0.44:0 (pp 0x60E105BC): advertise 45.0.0.0/8, label 39 (#36)
```

The following table describes the significant fields shown in the display.

Table 7: debug mpls ldp advertisements Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.
peer a.b.c.d:e	LDP identifier of the peer to which the advertisement was targeted.
(pp 0xnnnnnnnn)	Identifier for the data structure used to represent the peer at the label distribution level. Useful for correlating debug output.
advertise X	Identifies what was advertised to the peer--either an interface address (“a.b.c.d”) or label binding (“a.b.c.d/m, label t (#n)”).
(#n)	For a label binding advertisement, the sequence number of the label information base (LIB) modification that made it necessary to advertise the label.

Related Commands

Command	Description
debug mpls ldp bindings	Displays information about changes to the LIB used to keep track of label bindings learned from LDP peers through LDP downstream label distribution.

Command	Description
show mpls ip binding	Displays specified information about label bindings learned by LDP.
show mpls ldp neighbor	Displays the status of LDP sessions.

debug mpls ldp backoff

To display information about the label distribution protocol (LDP) backoff mechanism parameters, use the **debugmplsldpbackoff**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls ldp backoff
no debug mpls ldp backoff

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(10)ST	This command was introduced.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to monitor backoff parameters configured for LDP sessions.

Examples

The following shows sample output from the **debugmplsldpbackoff** command:

```
Router# debug mpls ldp backoff

LDP session establishment backoff debugging is on
Router#
Jan 6 22:31:13.012: ldp: Backoff peer ok: 12.12.12.12:0; backing off; threshold/count 8/6
Jan 6 22:31:13.824: ldp: Backoff peer ok: 12.12.12.12:1; backing off; threshold/count 8/6
Jan 6 22:31:17.848: ldp: Backoff peer ok: 12.12.12.12:0; backing off; threshold/count 8/6
Jan 6 22:31:18.220: ldp: Backoff peer ok: 12.12.12.12:1; backing off; threshold/count 8/6
Jan 6 22:31:21.908: ldp: Backoff peer ok: 12.12.12.12:0; backing off; threshold/count 8/6
Jan 6 22:31:22.980: ldp: Backoff peer ok: 12.12.12.12:1; backing off; threshold/count 8/6
Jan 6 22:31:25.724: ldp: Backoff peer ok: 12.12.12.12:0; backing off; threshold/count 8/7
Jan 6 22:31:26.944: ldp: Backoff peer ok: 12.12.12.12:1; backing off; threshold/count 8/7
Jan 6 22:31:30.140: ldp: Backoff peer ok: 12.12.12.12:0; backing off; threshold/count 8/7
Jan 6 22:31:31.932: ldp: Backoff peer ok: 12.12.12.12:1; backing off; threshold/count 8/7
Jan 6 22:31:35.028: ldp: Backoff peer ok: 12.12.12.12:0; backing off; threshold/count 8/7
Jan 6 22:31:35.788: ldp: Backoff peer ok: 12.12.12.12:1; backing off; threshold/count 8/7
Jan 6 22:31:39.332: ldp: Update backoff rec: 12.12.12.12:0, threshold = 8, tbl ents 2
Jan 6 22:31:39.640: ldp: Update backoff rec: 12.12.12.12:1, threshold = 8, tbl ents 2
```

The following table describes the significant fields shown in the display.

Table 8: debug mpls ldp backoff Field Descriptions

Field	Description
ldp	Identifies the Label Distribution Protocol.
Backoff peer ok: a.b.c.d:n	Identifies the LDP peer for which a session is being delayed because of a failure to establish a session due to incompatible configuration.
backing off;	Indicates that a session setup attempt failed and the LSR is delaying its next attempt (that is, is backing off).
threshold/count x/y	Identifies a set threshold (x) and a count (y) that represents the time that has passed since the last attempt to set up a session with the peer. The count is incremented every 15 seconds until it reaches the threshold. When the count equals the threshold, a fresh attempt is made to set up an LDP session with the peer.
Update backoff rec	Indicates that the backoff period is over and that it is time for another attempt to set up an LDP session.
threshold = x	Indicates the backoff time of x*15 seconds, for the next LDP session attempt with the peer.
tbl ents 2	Indicates unsuccessful attempts to set up an LDP session with two different LDP peers. In this example, attempts to set up sessions with LDP peers 12.12.12.12:0 and 12.12.12.12:1 are failing.

Related Commands

Command	Description
mpls ldp backoff	Configures session setup delay parameters for the LDP backoff mechanism.
show mpls ldp backoff	Displays information about the configured session setup backoff parameters and any potential LDP peers with which session setup attempts are being throttled.

debug mpls ldp bindings

To display information about addresses and label bindings learned from Label Distribution Protocol (LDP) peers by means of LDP downstream unsolicited label distribution, use the **debugmplsldpbindings**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls ldp bindings [**filter**] [**peer-acl** *acl*] [**prefix-acl** *acl*]
no debug mpls ldp bindings [**filter**] [**peer-acl** *acl*] [**prefix-acl** *acl*]

Syntax Description

filter	(Optional) Display information about LDP local label allocation filtering.
peer-acl *acl*	(Optional) Limits the displayed binding information to that learned from LDP peers permitted by the access control list (*acl*).
prefix-acl *acl*	(Optional) Limits the displayed binding information to that learned for prefixes permitted by the access control list (*acl*).

Command Default

Displays information about all bindings learned from all LDP peers.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to correspond to MPLS Internet Engineering Task Force (IETF) command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SRC	The **filter** keyword was added and the output of the command was updated to display information about LDP local label allocation filtering.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Usage Guidelines

Use this command to monitor label bindings and label switch router (LSR) addresses learned from LDP peers.

Note This command monitors non-LC-ATM labels (generic labels) only. Use the debug mpls atm-ldp states command to monitor LC-ATM activity.

Examples

The following is sample output from the **debugmplsldpbindings**command:

```
Router# debug mpls ldp bindings
tagcon:tibent(10.34.0.0/8):created; find route tags request
tagcon:tibent(10.34.0.0/8):label 3 (#2) assigned
tagcon:tibent(10.0.7.7/32):created; find route tags request
tagcon:tibent(10.0.7.7/32):label 24 (#4) assigned
tagcon:tibent(10.0.0.44/32):created; find route tags request
tagcon:tibent(10.0.0.44/32):label 33 (#30) assigned
tagcon:tibent(10.106.0.0/8):created; find route tags request
tagcon:tibent(10.106.0.0/8):label 34 (#32) assigned
tagcon:tibent(10.0.0.33/32):created; find route tags request
tagcon:tibent(10.0.0.33/32):label 3 (#34) assigned
tagcon:tibent(10.45.0.0/8):created; find route tags request
tagcon:tibent(10.45.0.0/8):label 39 (#36) assigned
tagcon:Assign peer id; 10.0.0.44:0:id 0
tagcon:10.0.0.44:0:10.0.0.44 added to addr<->ldp ident map
tagcon:10.0.0.44:0:10.34.0.44 added to addr<->ldp ident map
tagcon:10.0.0.44:0:10.45.0.44 added to addr<->ldp ident map
tagcon:tibent(10.0.0.44/32):rem label 3 from 10.0.0.44:0 added
tagcon:tibent(10.34.0.0/8):label 3 from 10.0.0.44:0 added
tagcon:tibent(10.45.0.0/8):label 3 from 10.0.0.44:0 added
tagcon:tibent(10.107.0.0/8):created; remote label learned
tagcon:tibent(10.107.0.0/8):label 55 from 10.0.0.44:0 added
tagcon:tibent(10.0.7.7/32):label 209 from 10.0.0.44:0 added
tagcon:tibent(10.0.0.33/32):label 207 from 10.0.0.44:0 added
```

The following table describes the significant fields shown in the display.

Table 9: debug mpls ldp bindings Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.
tibent(network/mask)	Destination that has a label binding change.
created; reason	An LIB entry has been created for the specified destination for the indicated reason.

Field	Description
rem label ...	Describes a change to the label bindings for the specified destination. The change is for a label binding learned from the specified LDP peer.
lcl label ...	Describes a change to a locally assigned (incoming) label for the specified destination.
(#n)	Sequence number of the modification to the LIB corresponding to the local label change.
a.b.c.d:n: e.f.g.h added to addr<->ldp ident map	The address e.f.g.h has been added to the set of addresses associated with LDP identifier a.b.c.d:n.

The following is output from the **debugmplsldpbindings**command when LDP local label allocation filtering is configured:

```
Router# debug mpls ldp
%SYS-5-CONFIG_I: Configured from console by console
Router# debug mpls ldp bindings filter
LDP Local Label Filtering changes debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls ldp label
Router(config-ldp-lbl)# allocate global host-routes
Router(config-ldp-lbl)#
 LDP LLAF: Enqueued work item to walk tib for all tables
 LDP LLAF: Withdraw local label for 10.10.7.0
 LDP LLAF: Withdraw local label for 10.10.8.0
 LDP LLAF: Withdraw local label for 10.10.9.0
Router(config-ldp-lbl)#
 LDP LLAF: announce zero local and path labels: 10.10.7.0
 LDP LLAF: announce zero local and path labels: 10.10.8.0
 LDP LLAF: announce zero local and path labels: 10.10.9.0
Router(config-ldp-lbl)#
Router(config-ldp-lbl)# no allocate global host-routes

Router(config-ldp-lbl)#
 LDP LLAF: Enqueued work item to walk tib for all tables
 tib: get path labels: 10.1.1.1/32, tableid: 0, Et1/0, nh 10.10.7.2
 LDP LLAF: 10.1.1.1 accepted, absence of filtering config
 tagcon: announce labels for: 10.1.1.1/32; nh 10.10.7.2, Et1/0, inlabel 17, outlabel imp-null
 (from 10.1.1.1:0), get path labels
 tib: get path labels: 10.2.2.2/32, tableid: 0, Et2/0, nh 10.10.8.2
 LDP LLAF: 10.2.2.2 accepted, absence of filtering config
 tagcon: announce labels for: 10.2.2.2/32; nh 10.10.8.2, Et2/0, inlabel 16, outlabel imp-null
 (from 10.2.2.2:0), get path labels
 tib: get path labels: 10.10.7.0/24, tableid: 0, Et1/0, nh 0.0.0.0
 LDP LLAF: 10.10.7.0 accepted, absence of filtering config
 tagcon: tibent(10.10.7.0/24): label 1 (#20) assigned
 tagcon: announce labels for: 10.10.7.0/24; nh 0.0.0.0, Et1/0, inlabel imp-null, outlabel
unknown (from 0.0.0.0:0), get path labels
 tib: get path labels: 10.10.8.0/24, tableid: 0, Et2/0, nh 0.0.0.0
 LDP LLAF: 10.10.8.0 accepted, absence of filtering config
 tagcon: tibent(10.10.8.0/24): label 1 (#21) assigned
 tagcon: announce labels for: 10.10.8.0/24; nh 0.0.0.0, Et2/0, inlabel imp-null, outlabel
unknown (from 0.0.0.0:0), get path labels
 tib: get path labels: 10.10.9.0/24, tableid: 0, Et1/0, nh 10.10.7.2
 LDP LLAF: 10.10.9.0 accepted, absence of filtering config
 tagcon: tibent(10.10.9.0/24): label 22 (#22) assigned
```

```
 tagcon: announce labels for: 10.10.9.0/24; nh 10.10.7.2, Et1/0, inlabel 22, outlabel
imp-null (from 10.1.1.1:0), get path labels
.
.
.
Router(config-ldp-lbl)# no mpls ldp label
Router(config-ldp-lbl)# end
Router# no debug mpls ldp bindings filter
```

The following table describes the significant fields shown in the display.

Table 10: debug mpls ldp bindings Field Descriptions with LDP Local Label Allocation Filtering

Field	Description
LDP LLAF	Indicates that the messages apply to LDP local label allocation filtering.
Withdraw local label for 10.10.7.0	Prefix 10.10.7.0 is not in the global routing table. LDP withdraws the label and does not assign a local labels.
announce zero local and path labels: 10.10.7.0	LDP does not announce local and path label for prefix 10.10.7.0.
tagcon: announce labels for:	The label control subsystem announces the next hop (nh) and labels for the named prefix.
tib: get path labels:	LDP LIB searches for the routing and forwarding path for the named prefix.
LDP LLAF: 10.1.1.1 accepted;	LDP accepts the prefix. The prefix was found in the global table (or accepted by the prefix list, if a prefix list was named as a filter).
tibent(network/mask)	Destination that has a label binding change.

Related Commands

Command	Description
debug mpls atm-ldp states	Displays information about label virtual circuit (lvc) state transitions as they occur.
show mpls ldp bindings	Displays the contents of the LIB.

debug mpls ldp checkpoint through debug mwi relay events

debug mpls ldp checkpoint

Note Effective with Cisco IOS Release 12.2(33)SRA, the **debugmplsldpcheckpoint**command is replaced by the **debugmplsvpnha**command. See the **debugmplsvpnha**command for more information.

To enable the display of Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) checkpoint debugging information, use the **debugmplsldpcheckpoint**command in privileged EXEC mode. To disable the display of MPLS LDP checkpoint debugging information, use the **no** form of this command.

debug mpls ldp checkpoint
no debug mpls ldp checkpoint

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging of MPLS LDP checkpointing is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was replaced by the **debugmplsvpnha**command.

Usage Guidelines

The following examples show sample output from the debug mpls ldp checkpoint command:

Sample Output on the Active Route Processor or PRE

```
Router# debug mpls ldp checkpoint
LDP Checkpointing events and errors debugging is on
LDP-CF: 0:10.3.3.3/32,20:: checkpointing local binding
LDP-CF: 0:10.3.3.3/32,20:: changing checkpoint state from none to add-send
LDP-CF: 0:10.3.3.3/32,20:: changing checkpoint state from add-send to add-wait
LDP-CF: received CF send-ack
LDP-CF: 0:10.3.3.3/32,20:: changing checkpoint state from add-wait to added
```

Sample Output on the Backup Route Processor or PRE

```
Router# debug mpls ldp checkpoint
LDP-CF: received 16-byte CF message: client 28 [0], ver 1, type 1
LDP-CF: 0:10.3.3.3/32,20:: adding checkpointed local binding
```

The following table describes the significant field in the sample display.

Table 11: debug mpls ldp checkpoint Command Field Descriptions

Field	Description
0:10.3.3.3/32,20::	The table ID, prefix, prefix length, and label of the checkpointed label binding.

Related Commands

Command	Description
show mpls ldp checkpoint	Displays information about the LDP checkpoint system on the active Route Processor.

debug mpls ldp graceful-restart

To display debugging information for Multiprotocol (MPLS) Label Distribution Protocol (LDP) stateful switchover (SSO) nonstop forwarding (NSF) support and Graceful Restart, use the **debugmplsldpgraceful-restart**command in privileged EXEC mode. To disable the display of this debugging information, use the **no** form of this command.

debug mpls ldp graceful-restart
no debug mpls ldp graceful-restart

Syntax Description

This command has no arguments or keywords.

Command Default

The display of debugging information is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(29)S	This command was introduced.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

This command shows events and errors related to LDP Graceful Restart.

Examples

The following example shows sample output from the debug mpls ldp graceful-restart command. The output shows that a session was lost. The status message show the events that happen during recovery of the bindings.

```
Router# debug mpls ldp graceful-restart
LDP GR: GR session 10.110.0.10:0:: lost
LDP GR: down nbr 10.110.0.10:0:: created [1 total]
LDP GR: GR session 10.110.0.10:0:: bindings retained
LDP GR: down nbr 10.110.0.10:0:: added all 7 addresses [7 total]
LDP GR: down nbr 10.110.0.10:0:: state change (None -> Reconnect-Wait)
LDP GR: down nbr 10.110.0.10:0:: reconnect timer started [120000 msecs]
LDP GR: down nbr 10.110.0.10:0:: added to bindings task queue [1 entries]
LDP GR: searching for down nbr record (10.110.0.10:0, 10.2.0.10)
LDP GR: search for down nbr record (10.110.0.10:0, 10.2.0.10) returned 10.110.0.10:0
LDP GR: Added FT Sess TLV (Rconn 120000, Rcov 120000) to INIT msg to 10.110.0.10:0
LDP GR: Tagcon querying for up to 12 bindings update tasks
LDP GR: down nbr 10.110.0.10:0:: requesting bindings MARK for {10.110.0.10:0, 1}
LDP GR: down nbr 10.110.0.10:0:: removed from bindings task queue [0 entries]
LDP GR: Requesting 1 bindings update tasks [0 left in queue]
```

```
LDP GR: 10.1.0.0/8:: updating binding from 10.110.0.10:0, inst 1:: marking stale;
LDP GR: 10.2.0.0/16:: updating binding from 10.110.0.10:0, inst 1:: marking stale;
LDP GR: 10.0.0.14/32:: updating binding from 10.110.0.10:0, inst 1:: marking stale;
LDP GR: searching for down nbr record (10.110.0.10:0, 10.2.0.10)
LDP GR: search for down nbr record (10.110.0.10:0, 10.2.0.10) returned 10.110.0.10:0
LDP GR: Added FT Sess TLV (Rconn 120000, Rcov 120000) to INIT msg to 10.110.0.10:0
LDP GR: searching for down nbr record (10.110.0.10:0, 10.2.0.10)
LDP GR: search for down nbr record (10.110.0.10:0, 10.2.0.10) returned 10.110.0.10:0
LDP GR: Added FT Sess TLV (Rconn 120000, Rcov 120000) to INIT msg to 10.110.0.10:0
LDP GR: searching for down nbr record (10.110.0.10:0, 10.2.0.10)
LDP GR: search for down nbr record (10.110.0.10:0, 10.2.0.10) returned 10.110.0.10:0
LDP GR: Added FT Sess TLV (Rconn 120000, Rcov 120000) to INIT msg to 10.110.0.10:0
LDP GR: searching for down nbr record (10.110.0.10:0, 10.2.0.10)
LDP GR: search for down nbr record (10.110.0.10:0, 10.2.0.10) returned 10.110.0.10:0
LDP GR: Added FT Sess TLV (Rconn 120000, Rcov 120000) to INIT msg to 10.110.0.10:0
LDP GR: searching for down nbr record (10.110.0.10:0, 10.2.0.10)
LDP GR: search for down nbr record (10.110.0.10:0, 10.2.0.10) returned 10.110.0.10:0
LDP GR: Added FT Sess TLV (Rconn 120000, Rcov 120000) to INIT msg to 10.110.0.10:0
LDP GR: Received FT Sess TLV from 10.110.0.10:0  (fl 0x1, rs 0x0, rconn 120000, rcov 120000)
LDP GR: GR session 10.110.0.10:0:: allocated instance, 2
LDP GR: GR session 10.110.0.10:0:: established
LDP GR: GR session 10.110.0.10:0:: found down nbr 10.110.0.10:0
LDP GR: down nbr 10.110.0.10:0:: reconnect timer stopped
LDP GR: down nbr 10.110.0.10:0:: state change (Reconnect-Wait -> Recovering)
LDP GR: down nbr 10.110.0.10:0:: recovery timer started [120000 msecs]
%LDP-5-GR: GR session 10.110.0.10:0 (inst. 2): starting graceful recovery
%LDP-5-NBRCHG: LDP Neighbor 10.110.0.10:0 is UP
LDP GR: 10.1.0.0//8:: refreshing stale binding from 10.110.0.10:0, inst 1 -> inst 2
LDP GR: 10.43.0.0//16:: refreshing stale binding from 10.110.0.10:0, inst 1 -> inst 2
LDP GR: down nbr 10.110.0.10:0:: recovery timer expired
%LDP-5-GR: GR session 10.110.0.10:0 (inst. 2): completed graceful recovery
LDP GR: down nbr 10.110.0.10:0:: destroying record [0 left]
LDP GR: down nbr 10.110.0.10:0:: state change (Recovering -> Delete-Wait)
LDP GR: down nbr 10.110.0.10:0:: added to bindings task queue [1 entries]
LDP GR: Tagcon querying for up to 12 bindings update tasks
LDP GR: down nbr 10.110.0.10:0:: requesting bindings DEL for {10.110.0.10:0, 1}
LDP GR: down nbr 10.110.0.10:0:: removed from bindings task queue [0 entries]
LDP GR: Requesting 1 bindings update tasks [0 left in queue]
LDP GR: GR session 10.110.0.10:0:: released instance, 1
```

The debug output is formatted in three general ways.

- LDP GR: GR session 10.110.0.10:0:: found down nbr 10.110.0.10:0
- down nbr 10.110.0.10:0:: removed from bindings task queue [0 entries]
- LDP GR: 2.0.0.0/8:: updating binding from 10.110.0.10:0, inst 1:: marking stale;

The following table describes the fields for the debug command output.

Table 12: debug mpls ldp graceful-restart Command Field Descriptions

Field	Description
LDP GR	Identifies LDP Graceful Restart application
GR session 10.110.0.10:0	ID of the LDP session that is enabled for Graceful Restart.
found down nbr 10.110.0.10:0	Describes the event that is happening to that LDP session.
down nbr 10.110.0.10:0::	Identifies the Down Neighbor record, which logs the state of a recently lost Graceful Restart session.

Field	Description
removed from bindings task queue [0 entries]	Describes the event that is happening to the recently lost Graceful Restart session.
2.0.0.0/8::	Identifies the Forwarding Equivalence Class (FEC) associated with the remote label binding being modified. The FEC identifies the Label Information Base (LIB) entry.
updating binding	Lists the operation being performed on the remote label binding.
10.110.0.10:0, inst 1:: marking stale;	Identifies the LDP session during which the remote label binding was learned.

Related Commands

Command	Description
show mpls ldp graceful-restart	Displays a summary of the LDP Graceful Restart status.

debug mpls ldp igp sync

To enable the display of events related to the Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP)-Interior Gateway Protocol (IGP) Synchronization feature, use the **debugmplsldpigpsync**command in **privilegedEXEC**mode. To disable this feature, use the **no** form of this command.

debug mpls ldp igp sync [**interface** *interface*] [**peer** *acl*]
no debug mpls ldp igp sync [**interface** *interface*] [**peer** *acl*]

Syntax Description

interface *interface*	(Optional) Enables the display of MPLS LDP-IGP synchronization events for the specified interface.
peer *acl*	(Optional) Enables the display of MPLS LDP-IGP synchronization events for the specified peer access control list (ACL).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(30)S	This command was introduced.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.
12.0(32)S	The command output was modified to display events related to the delay timer on interfaces running Open Shortest Path First (OSPF) processes, if the delay timer is configured.
12.0(32)SY	The command output was modified to display events related to synchronization on interfaces running Intermediate System-to-Intermediate System (IS-IS) processes.
12.4(12)	This command was integrated into Cisco IOS Release 12.4(12).
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following example shows events related to MPLS LDP-IGP synchronization on interfaces running OSPF:

```
Router# debug mpls ldp igp sync
LDP-SYNC: Et0/0, OSPF 1: notify status (required, not achieved, no delay, holddown infinite)
 internal status (achieved, timer running)
LDP-SYNC: E1/0, 10.0.0.1: Adj being deleted, sync_achieved goes down
LDP-SYNC: E1/0, OSPF 1: notify status (required, not achieved, delay, holddown infinite)
LDP-SYNC: Et0/0: Session already up and interface address advertised, sync_achieved comes
up.
LDP-SYNC: Et0/0, OSPF 1: notify status (required, achieved, no delay, holddown infinite)
```

The following example shows events associated when an IS-IS instance, ISIS-1, is configured for synchronization:

```
Router# debug mpls ldp igp sync
07:59:27: LDP-SYNC: Et0/0, OSPF 1: notify status (required, not achieved, no delay, holddown
 infinite) internal status (achieved, timer running)
07:59:27: LDP-SYNC: Enqueue request req_type 0 IGP ISIS ISIS-1 interface none.
07:59:27: LDP-SYNC: ISIS ISIS-1: SYNC enabled, added to global tree, informed IGP.
07:59:27: LDP-SYNC: Enqueue request req_type 3 IGP ISIS ISIS-1 interface Et0/0.
07:59:27: LDP-SYNC: Enqueue request req_type 3 IGP ISIS ISIS-1 interface Et0/0.
07:59:27: LDP-SYNC: Et0/0, ISIS ISIS-1: Added to per-interface IGP list.
07:59:27: LDP-SYNC: Et0/0: Enabled for SYNC by IGP
07:59:27: LDP-SYNC: Et0/0, ISIS ISIS-1: notify status (required, not achieved, delay,
holddown infinite)
07:59:27: LDP-SYNC: Et0/0, ISIS ISIS-1: Ignore IGP enable-interface request: already enabled.
```

The following table describes the significant fields shown in the displays.

Table 13: debug mpls ldp igp sync Field Descriptions

Field	Description
sync_achieved	The first line of the output for an interface shows the status of the MPLS LDP-IGP Synchronization feature in relation to the status of the interface.
notify status	Notify status shows the following MPLS LDP-IGP synchronization information for each interface: • If MPLS LDP-IGP synchronization is required. • If MPLS LDP-IGP synchronization has been achieved. • If the IGP should wait for MPLS LDP-IGP synchronization to be achieved. • The length of time the IGP should wait for MPLS LDP-IGP synchronization to be achieved.
internal status	Internal status displays the LDP internal synchronization status and the state of the timer. The internal status can be achieved or not achieved. The timer state can be running or not running.

The following example shows events associated with MPLS LDP-IGP synchronization on interfaces running OSPF when you configured a delay timer:

```
Router# debug mpls ldp igp sync
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: notify status (required, not achieved, no
delay, holddown infinite) internal status (achieved, timer running)
!
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: Sync disabled by IGP. Stop delay timer
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: TAGSW subblock destroyed. Stop delay timer
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: Sync down. Stop delay timer
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: Delay notifying IGP of sync achieved for 60
seconds
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: Delay timer expired, notify IGP of sync
achieved
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: Delay timer expired but sync is no longer
required won't notify IGP of sync achieved
*Jan 3 04:38:49.571: LDP-SYNC: Et0/0, OSPF 1: Delay timer expired but sync is down won't
notify IGP of sync achieved
```

Related Commands

Command	Description
mpls ldp sync	Enables MPLS LDP-IGP synchronization on all interfaces that belong to an OSPF process or IS-IS process.
show mpls ldp igp sync	Displays the status of the MPLS LDP-IGP synchronization process.

debug mpls ldp messages

To display specific information (such as message type, source, and destination) about Label Distribution Protocol (LDP) messages sent to and received from LDP peers, use the **debugmplsldpmessages**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls ldp messages {**sent** | **received**} [**all**] [**peer-acl** *acl*]
no debug mpls ldp messages {**sent** | **received**} [**all**] [**peer-acl** *acl*]

Syntax Description

sent	Displays LDP messages sent to LDP peers permitted by the access control list (*ACL)*.
received	Displays LDP messages received from LDP peers permitted by the *ACL*.
all	(Optional) Displays all LDP messages sent to and received from LDP peers (including periodic keepalive messages) permitted by the ACL.
peer-acl *acl*	(Optional) Limits the messages displayed for LDP peers in accordance with the ACL.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(50)SY	This command was integrated into Cisco IOS Release 12.2(50)SY.

Usage Guidelines

LDP requires periodic transmission of keepalive messages. If you do not specify the **all** keyword, periodic keepalive messages are not displayed.

Examples

The following is sample output from the **debugmplsldpmessages** command:

```
Router# debug mpls ldp messages received
LDP received messages, excluding periodic Keep Alives debugging is on
Router# debug mpls ldp messages sent
LDP sent PDUs, excluding periodic Keep Alives debugging is on
ldp: Rcvd init msg from 192.168.10.1 (pp 0x0)
ldp: Sent init msg to 192.168.10.1:0 (pp 0x0)
```

```
ldp: Sent keepalive msg to 192.168.10.1:0 (pp 0x0)
ldp: Rcvd keepalive msg from 192.168.10.1:0 (pp 0x0)
ldp: Sent address msg to 192.168.10.1:0 (pp 0x610F00E0)
ldp: Sent label mapping msg to 192.168.10.1:0 (pp 0x610F00E0)
ldp: Sent label mapping msg to 192.168.10.1:0 (pp 0x610F00E0)
ldp: Sent label mapping msg to 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd address msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610F00E0)
```

The following table describes the significant fields shown in the display.

Table 14: debug mpls ldp messages Field Descriptions

Field	Description
ldp:	Identifies the source of the displayed information as LDP.
Rcvd xxx msg Sent xxx msg	Type of message received or sent.
from a.b.c.d	Host that sent the message. Used in the early stages of the opening of an LDP session, when the LDP identifier is not yet known.
from a.b.c.d:e to a.b.c.d:e	LDP identifier of the peer that sent the message or to which the message was sent.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer at the label distribution level. Useful for correlating debug output.

Related Commands

Command	Description
debug mpls ldp session io	Displays the contents of LDP messages sent to and received from LDP peers.

debug mpls ldp nsr

To enable the display of Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) nonstop routing (NSR) debugging events for all NSR sessions or for a specified peer, use the **debug mpls ldp nsr** command in privileged EXEC mode. To disable the display of MPLS LDP NSR debugging information, use the **no** form of this command.

debug mpls ldp nsr [**peer-acl** *acl-name*]
no debug mpls ldp nsr [**peer-acl** *acl-name*]

Syntax Description

peer-acl *acl-name*	(Optional) Displays LDP NSR events for the specified peer access list.

Command Default

Debugging of MPLS LDP NSR events are not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.9S	This command was introduced.

Example

The following is sample output from the **debug mpls ldp nsr** command:

```
Device# debug mpls ldp nsr
*Feb  5 22:14:55.666: LDP NSR is enabled
*Feb  5 22:14:55.666: LDP Non-Stop-Routing has been enabled
*Feb  5 22:14:55.871: LDP-CF: 0:0x2A9B99C9B8 for Serial4/0, adj_addr/xport_addr
10.2.4.4/10.4.0.1:: received standby session-up, 9, in state init-sent
*Feb  5 22:14:55.871: LDP NSR: Sess Sync Record created for peer 10.4.0.1:0, inst 2,type 1
*Feb  5 22:14:55.871: LDP NSR: Addr sync Rec added to tree for peer 10.4.0.1:0, inst 2,
msg-id 0, num-rec 3
*Feb  5 22:14:55.871: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from none
to send for msgid 0
*Feb  5 22:14:55.871: LDP NSR: Sess Sync Record created for peer 10.4.0.1:0, inst 2,type 3
*Feb  5 22:14:55.871: LDP NSR: Session Sync record deleted for peer 10.4.0.1:0, inst 2,
type 3
*Feb  5 22:14:55.871: LDP NSR: Sess Sync Record created for peer 10.4.0.1:0, inst 2,type 2
*Feb  5 22:14:55.872: LDP NSR: Rbind sync Rec added to tree for peer 10.4.0.1:0, inst 2,
msg-id 2, num-rec 9
*Feb  5 22:14:55.872: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from none
to send for msgid 2
*Feb  5 22:14:55.872: LDP NSR: Sess Sync Record created for peer 10.4.0.1:0, inst 2,type 4
*Feb  5 22:14:55.872: LDP NSR: Cap sync Rec added to tree for peer 10.4.0.1:0, inst 2,
msg-id 3, num-rec 10
*Feb  5 22:14:55.872: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from none
to send for msgid 3
*Feb  5 22:14:55.872: LDP NSR: Sess Sync Addr Msg for Peer 10.4.0.1:0, inst 2, msg_id 0,
num_records 3
*Feb  5 22:14:55.872: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from send
to wait for msgid 0
*Feb  5 22:14:55.872: LDP NSR: Peer 10.4.0.1:0 Addr Session sync sent, action 9, state wait
*Feb  5 22:14:55.872: LDP NSR: Sess Sync Rbind Msg for Peer 10.4.0.1:0, msg_id 2, num_records
 9
*Feb  5 22:14:55.872: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from send
```

```
to wait for msgid 2
*Feb  5 22:14:55.872: LDP NSR: Peer 10.4.0.1:0 Session sync sent, action 11, state wait
*Feb  5 22:14:55.872: LDP NSR: Sess Sync Cap Msg for Peer 10.4.0.1:0, msg_id 3, num_records
 10
*Feb  5 22:14:55.872: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from send
to wait for msgid 3
*Feb  5 22:14:55.872: LDP NSR: Peer 10.4.0.1:0 Session sync sent, action 12, state wait
*Feb  5 22:14:55.873: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from wait
to none for msgid 0
*Feb  5 22:14:55.873: LDP NSR: Session Sync record deleted for peer 10.4.0.1:0, inst 2,
type 1
*Feb  5 22:14:55.873: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from wait
to none for msgid 2
*Feb  5 22:14:55.873: LDP NSR: Session Sync record deleted for peer 10.4.0.1:0, inst 2,
type 2
*Feb  5 22:14:56.488: LDP NSR: Peer10.4.0.1:0, Inst 2, Changing sync_rec state from wait
to none for msgid 3
*Feb  5 22:14:56.488: LDP NSR: Session Sync record deleted for peer 10.4.0.1:0, inst 2,
type 4
*Feb  5 22:14:56.488: LDP-CF: 0:0x2A9B99C9B8 for Serial4/0, adj_addr/xport_addr
10.2.4.4/10.4.0.1:: received Session Sync Done, 13, in state session-sync
*Feb  5 22:14:56.488: LDP NSR: Active Chkpt sess_sync_done for Peer 10.4.0.1:0, inst 2,
type 11, seq 11
```

Related Commands

Command	Description
mpls ldp nsr	Enables or disables NSR for LDP sessions.

debug mpls ldp peer state-machine

To display information about state transitions for label distribution protocol (LDP) sessions, use the **debugmplsldppeerstate-machine**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls ldp peer state-machine
no debug mpls ldp peer state-machine

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

LDP manages peer sessions by means of two coupled state machines:

- A low-level state machine that deals with session establishment and shutdown
- A high-level state machine that deals with setting up and shutting down label advertisement

Use the **debugmplsldpsessionstate-machine** command to monitor the lower-level session state machine.

Use the **debugmplsldppeerstate-machine** command to monitor the higher-level session state machine.

Examples

The following shows sample output from the **debugmplsldppeerstate-machine** command:

```
Router# debug mpls ldp peer state-machine
tagcon: start session TCP timers for 144.0.0.44:0 (pp 0x610EEC84)
tagcon: Enqueue peer up work for 144.0.0.44:0 (pp 0x610EEC84)
tagcon: peer 144.0.0.44:0 (pp 0x610EEC84): Event unsol open
    unsol op pdg -> estab
tagcon: Send initial advertisements to peer 144.0.0.44:0
tagcon: Initial address advertisement to peer 144.0.0.44:0
tagcon: Initial label advertisement to peer 144.0.0.44:0
...
tagcon: peer 144.0.0.44:0 (pp 0x610EEC84): Event down
    estab -> destroyed
tagcon: peer 144.0.0.44:0 (pp 0x610EEC84): Event cleanup done
    destroyed -> non-ex
```

The following table describes the significant fields shown in the display.

Table 15: debug mpls ldp peer state-machine Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.
a.b.c.d:e	LDP identifier of the peer for the session with the state change.
(pp 0xnnnnnnnn)	Address of the data structure used to represent the peer at the label distribution level. This address is useful for correlating debug output.
Event *E*	Event causing the state change.
s1 -> s2	State of the LDP session has changed from state s1 to state s2.

Related Commands

Command	Description
debug mpls ldp session io	Displays information about LDP messages sent to or received from LDP peers.
show mpls ldp neighbor	Displays the status of LDP sessions.

debug mpls ldp prev-label

To display debug information when a local label binding associated with a prefix is withdrawn and freed, use the **debugmplsldpprev-label**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls ldp prev-label [**prefix-acl** *acl* [**peer-acl** *acl*]]
no debug mpls ldp prev-label [**prefix-acl** *acl* [**peer-acl** *acl*]]

Syntax Description

prefix-acl *acl*	(Optional) Limits the displayed binding information to that allocated for prefixes permitted by a prefix access control list (ACL).
peer-acl *acl*	(Optional) Limits the displayed binding withdraw information to those Label Distribution Protocol (LDP) peers permitted by a peer ACL.

Command Default

Debugging of previous local label binding changes is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(21)ST	This command was introduced.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.

Usage Guidelines

Use this command to monitor LDP information when a local label binding associated with a prefix is withdrawn and freed. LDP withdraws a previously advertised label before advertising a new label.

If you enter the **debugmplsldpprev-label**command without an optional keyword and argument, the command displays output for all previous label binding changes. Use the **prefix-acl***acl*or**peer-acl***acl*keywords and arguments to limit the output to prefixes defined by the respective ACLs.

Examples

The following is sample output from the **debugmplsldpprev-label** command:

```
Router# debug mpls ldp prev-label
tagcon: Changing state to WITHDRAWN for prefix=10.0.1.1, label31
tagcon: Creating prev_lbl_info for prefix=10.0.1.1, label31
tagcon: noroute hold timer expired for 10.0.1.1/255.255.255.255, tag withdrawn, seqno 47
tagcon: tibent(10.0.1.1/32): label 32 from 10.0.0.2:0 removed
tagcon: Deleting prev label info for prefix = 10.0.1.1, tag = 31
```

The following table describes the significant fields shown in the display.

Table 16: debug mpls ldp prev-label Field Descriptions

Field	Description
tagcon:	Identifies the source of the message as the label control subsystem.

Field	Description
Changing state to WITHDRAWN	Describes the label binding change; in this case, the label is to be withdrawn.
for prefix=10.0.1.1	The prefix (10.0.1.1) from which the local label binding is to be withdrawn and freed.
label31	The local label binding (31) that is to be withdrawn from the prefix.
tibent(10.0.1.1/32)	The hostname, network, and mask for the destination that has a label binding change.

Related Commands

Command	Description
debug mpls ldp bindings	Displays information about addresses and label bindings learned from LDP peers by means of LDP downstream unsolicited label distribution.

debug mpls ldp session io

To display the contents of Label Distribution Protocol (LDP) messages sent to and received from LDP peers, use the **debugmplsldpsessionio**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls ldp session io [**all**] [**peer-acl** *acl*]
no debug mpls ldp session io [**all**] [**peer-acl** *acl*]

Syntax Description

all	(Optional) Includes the contents of periodic keepalive messages in the displayed message output to LDP peers.
peer-acl *acl*	(Optional) Limits the displayed message output to the LDP peers permitted by the access control list (*ACL*).

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(50)SY	This command was integrated into Cisco IOS Release 12.2(50)SY.

Usage Guidelines

Displays the contents of all messages sent and received, except for periodic keepalive messages.

Examples

The following is sample output from the **debugmplsldpsessionio** command:

```
Router# debug mpls ldp session io all
LDP session I/O, including periodic Keep Alives debugging is on
Router# debug mpls ldp session io peer-acl acl1
LDP session I/O, excluding periodic Keep Alives debugging is on for peer ACL acl1
ldp: Rcvd init msg from 192.168.10.1 (pp 0x0)
ldp: LDP init msg: PDU hdr: LDP Id: 192.168.10.1:0; Msg Contents:
  0x00 0x01 0x00 0x20 0x90 0x00 0x00 0x2C 0x00 0x00 0x02 0x00 0x00 0x16 0x00 0x00
  0x10 0x21 0x05 0x00 0x00 0x0E 0x00 0x01 0x00 0xB4 0x00 0x00 0x00 0x00 0x85 0x00
  0x00 0x21 0x00 0x00
ldp: Sent init msg to 192.168.10.1:0 (pp 0x0)
```

```
ldp: LDP init msg: PDU hdr: LDP Id: 192.168.10.2:0; Msg Contents:
  0x00 0x01 0x00 0x20 0x85 0x00 0x00 0x21 0x00 0x00 0x02 0x00 0x00 0x16 0x00 0x00
  0x06 0x32 0x05 0x00 0x00 0x0E 0x00 0x01 0x00 0xB4 0x00 0x00 0x00 0x00 0x90 0x00
  0x00 0x2C 0x00 0x00
ldp: Sent keepalive msg to 192.168.10.1:0 (pp 0x0)
ldp: LDP keepalive msg: PDU hdr: LDP Id: 192.168.10.2:0; Msg Contents:
  0x00 0x01 0x00 0x0E 0x85 0x00 0x00 0x21 0x00 0x00 0x02 0x01 0x00 0x04 0x00 0x00
  0x06 0x33
ldp: Rcvd keepalive msg from 192.168.10.1:0 (pp 0x0)
ldp: LDP keepalive msg: PDU hdr: LDP Id: 192.168.10.1:0; Msg Contents:
  0x00 0x01 0x00 0x0E 0x90 0x00 0x00 0x2C 0x00 0x00 0x02 0x01 0x00 0x04 0x00 0x00
  0x10 0x22
ldp: Sent address msg to 192.168.10.1:0 (pp 0x610ECDD0)
ldp: LDP address msg: PDU hdr: LDP Id: 192.168.10.2:0; Msg Contents:
  0x00 0x01 0x00 0x34 0x85 0x00 0x00 0x21 0x00 0x00 0x03 0x00 0x00 0x2A 0x00 0x00
  0x06 0x34 0x01 0x01 0x00 0x22 0x00 0x01 0x02 0x00 0x00 0xA3 0x82 0x42 0x00 0x21
  0x82 0x4D 0x00 0x21 0x85 0x00 0x00 0x21 0x22 0x00 0x00 0x21 0x67 0x00 0x00 0x21
  0x23 0x00 0x00 0x21 0x26 0x00 0x00 0x21
ldp: Sent label mapping msg to 192.168.10.1:0 (pp 0x610ECDD0)
ldp: LDP label mapping msg: PDU hdr: LDP Id: 192.168.10.2:0; Msg Contents:
  0x00 0x01 0x00 0x22 0x85 0x00 0x00 0x21 0x00 0x00 0x04 0x00 0x00 0x18 0x00 0x00
  0x06 0x36 0x01 0x00 0x00 0x08 0x02 0x00 0x01 0x20 0xCB 0x00 0x07 0x07 0x02 0x00
  0x00 0x04 0x00 0x00 0x00 0x18
ldp: Rcvd address msg from 192.168.10.1:0 (pp 0x610ECDD0)
ldp: LDP address msg: PDU hdr: LDP Id: 192.168.10.1:0; Msg Contents:
  0x00 0x01 0x00 0x24 0x90 0x00 0x00 0x2C 0x00 0x00 0x03 0x00 0x00 0x1A 0x00 0x00
  0x10 0x23 0x01 0x01 0x00 0x12 0x00 0x01 0x90 0x00 0x00 0x2C 0x02 0x00 0x00 0xA4
  0x22 0x00 0x00 0x2C 0x2D 0x00 0x00 0x2C
ldp: Rcvd label mapping msg from 192.168.10.1:0 (pp 0x610ECDD0)
ldp: LDP label mapping msg: PDU hdr: LDP Id: 192.168.10.1:0; Msg Contents:
  0x00 0x01 0x00 0x22 0x90 0x00 0x00 0x2C 0x00 0x00 0x04 0x00 0x00 0x18 0x00 0x00
  0x10 0x24 0x01 0x00 0x00 0x08 0x02 0x00 0x01 0x20 0x90 0x00 0x00 0x2C 0x02 0x00
  0x00 0x04 0x00 0x00 0x00 0x03
```

The following table describes the significant fields shown in the display.

Table 17: debug mpls ldp session io Field Descriptions

Field	Description
ldp:	Identifies the source of the message as LDP.
Rcvd xxx msg	Indicates that a message of the specified type has been received.
from a.b.c.d	Host to which the message has been sent. Used in the early stages of the opening of an LDP session when the LDP identifier is not yet known.
Sent xxx msg	Indicates that a message of the specified type has been sent.
to a.b.c.d	Host to which the message has been sent. Used in the early stages of the opening of an LDP session when the LDP identifier is not yet known.
to a.b.c.d:e	LDP identifier of the peer to which the message has been sent.
(pp 0xnnnnnnnn)	Identifies the data structure used to represent the peer at the label distribution level. Useful for correlating debug output.
LDP xxx msg	Type of message that has been sent.

Field	Description
PDU hdr: LDP Id: a.b.c.d:e	LDP identifier of the sender included in the LDP protocol data unit (PDU) header.
Msg contents: 0xnn ... 0xnn	Contents of the message represented as a sequence of bytes.

Related Commands

Command	Description
debug mpls ldp messages	Displays specific information (such as message type, source, and destination) regarding LDP messages sent to and received from LDP peers.
debug mpls ldp session state-machine	Displays information about state transitions for LDP sessions.

debug mpls ldp session protection

To enable the display of events related to MPLS LDP Session Protection, use the **debug mpls ldp session protection**command in privileged EXEC mode. To disable this feature, use the **no** form of this command.

debug mpls ldp session protection [**peer-acl** *acl*]
no debug mpls ldp session protection [**peer-acl** *acl*]

Syntax Description

peer-acl acl	(Optional) Enables the display of events for the peers whose router IDs are listed in the access control list.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(30)S	This command was introduced.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

In the following example, the display of events related to MPLS LDP Session Protection are enabled:

```
Router# debug mpls ldp session protection
```

Related Commands

Command	Description
clear mpls ldp neighbor	Forcibly resets an LDP session.
show mpls ldp neighbor	Displays the contents of the LDP.

debug mpls ldp session state-machine

To display information about state transitions for label distribution protocol (LDP) sessions, use the **debugmplsldpsessionstate-machine**command in privileged EXEC mode. To disable this feature, use the no form of this command.

debug mpls ldp session state-machine [**peer-acl** *acl*]
no debug mpls ldp session state-machine [**peer-acl** *acl*]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed information to that for LDP peers permitted by the access control list (*acl*).

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

LDP manages peer sessions by means of two coupled-state machines:

- A low-level state machine that deals with session establishment and shutdown
- A high-level state machine that deals with setting up and shutting down label advertisement

Use the **debugmplsldpsessionstate-machine** command to monitor the lower-level session state machine.

Use the **debugmplsldppeerstate-machine** command to monitor the higher-level session state machine.

Examples

The following shows sample output from the **debugmplsldpsessionstate-machine** command:

```
Router# debug mpls ldp session state-machine
ldp: ptcl_adj:144.0.0.44(0x610EED30): Non-existent -> Role pasv
ldp: create ptcl_adj: tp = 0x610EED30, ipaddr = 144.0.0.44
ldp: ptcl_adj:144.0.0.44(0x610EED30): Event: Xport opened;
    Role pasv -> Role pasv
ldp: ptcl_adj:34.0.0.44(0x610EED30): Event: Rcv Init;
    Role pasv -> Init rcvd pasv
ldp: ptcl_adj:34.0.0.44(0x610EED30): Event: Rcv KA;
    Init rcvd pasv -> Oper
ldp: ptcl_adj:unknown(0x610EED30): Event: Xport closed;
    Oper -> Non-existent
```

The following table describes the significant fields shown in the display.

Table 18: debug mpls ldp session state-machine Field Descriptions

Field	Description
ldp:	Identifies the source of the message as LDP.
ptcl_adj:a.b.c.d	Identifies the network address of the LDP peer.
(0xnnnnnnnn)	Identifies the data structure used to represent the peer at the protocol level. Useful for correlating debug output.
Event: *E*	Event that caused the state transition.
s1 -> s2	State of the LDP session has changed from state s1 to state s2.

Related Commands

Command	Description
debug mpls ldp peer state-machine	Displays information about state transitions for LDP sessions.

debug mpls ldp targeted-neighbors

To display information about the target neighbor mechanism, use the **debugmplsldptargeted-neighbors**command in privileged EXEC mode. This mechanism establishes label distribution protocol (LDP) adjacencies to peers that are not directly adjacent, such as peers at either end of a tunnel. To disable this feature, use the no form of this command.

debug mpls ldp targeted-neighbors
no debug mpls ldp targeted-neighbors

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Platforms that are not directly connected may engage in LDP label distribution (for example, to support two-level labeling across an LSP tunnel).

An LDP session between nondirectly connected label switch routers (LSRs) is called a targeted session and is supported by LDP extended discovery which uses targeted Hello messages sent to specific IP addresses.

An LSR (Router 1) attempting to initiate an LDP targeted session with another LSR (Router 2) sends targeted Hello messages sent to a specific IP address of Router 2. If the configuration of Router 2 permits it to respond to targeted Hello messages from Router 1, it does so, and the LDP session can be established. In this situation, Router 1 is said to be an active LSR for the targeted session because it initiated the targeted Hello messages; Router 2 is said to be a passive LSR for the session because it responded to them.

As with LDP sessions between two directly connected LSRs, it is possible for a targeted session to be the result of multiple discovery activities which are targeted to different IP addresses for the same LSR. In addition, it is possible for both LSRs in a targeted session to be active and for both to be passive.

The debug messages enabled by debug mpls ldp targeted-neighbors report activity relating to targeted sessions.

Examples

The following shows sample output from the **debugmplsldptargeted-neighbors**command:

```
Router# debug mpls ldp targeted-neighbors
ldp-trgtnbr: 144.0.0.44 Req active
ldp-trgtnbr: 144.0.0.44 allocated
ldp-trgtnbr: 144.0.0.44 Set peer start; flags 0x0
ldp-trgtnbr: 144.0.0.44 Defer peer cleanup; cleancnt 1
ldp-trgtnbr: 144.0.0.44 Set peer finished; flags 0xF
ldp-trgtnbr: 144.0.0.44 ref count incremented to 1
ldp-trgtnbr: 144.0.0.44 Release active; ref count decremented to 0
ldp-trgtnbr: 144.0.0.44 Clear peer start; flags 0xF
ldp-trgtnbr: 144.0.0.44 Undefer cleanup start; clearcnt 0, flags 0xC
ldp-trgtnbr: 144.0.0.44 Undefer cleanup finish; clearcnt 0, flags 0x8
ldp-trgtnbr: 144.0.0.44 Clear peer finished; flags 0x8
ldp-trgtnbr: 144.0.0.44 freed
```

The following table describes the significant fields shown in the display.

Table 19: debug mpls ldp targeted-neighbors Field Descriptions

Field	Description
ldp-trgtnbr:	Identifies this as an LDP targeted neighbor debug statement.
144.0.0.44	IP address for the targeted neighbor.

Related Commands

Command	Description
show mpls ldp neighbor	Displays the status of LDP protocol sessions.

debug mpls ldp transport connections

To display information about the Transmission Control Protocol (TCP) connections used to support label distribution protocol (LDP) sessions, use the **debugmplsldptransportconnections**command in privilegedEXEC mode. To disable this feature, use the no form of this command.

debug mpls ldp transport connections [**peer-acl** *acl*] [**interface** *interface*]
no debug mpls ldp transport connections [**peer-acl** *acl*] [**interface** *interface*]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed information to that for LDP peers permitted by the access control list (*acl*).
interface *interface*	(Optional) Limits the displayed information to that for the specified interface.

Command Default

Display information about LDP TCP connection activity for all peers and all interfaces.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to monitor LDP activity relating to the establishment of the transport (TCP) connection for LDP sessions.

When two devices establish a TCP connection for an LDP session, the device with the larger transport address plays an active role and the other plays a passive role. The active device attempts to establish a TCP connection to the well-known LDP port at the passive device. The passive device waits for the connection to the well-known port to be established.

Examples

The following shows sample output from the **debugmplsldptransportconnections** command:

```
Router#
debug mpls ldp transport connections
Debug output at active peer:
ldp: Opening listen port 646 for 144.0.0.44, 34.0.0.44
ldp: Open LDP listen TCB 0x60E105BC; lport = 646; fhost = 144.0.0.44
ldp: Add listen TCB to list; tcb 0x60E105BC; addr 144.0.0.44
ldp: Incoming ldp conn 133.0.0.33:646 <-> 144.0.0.44:11042
ldp: create ptcl_adj: tp = 0x610ECD64, ipaddr = 144.0.0.44
Debug output at passive peer:
ldp: Opening ldp conn; adj 0x60BAC33C, 144.0.0.44 <-> 133.0.0.33
ldp: ldp conn is up; adj 0x60BAC33C, 144.0.0.44:11042 <-> 133.0.0.33:646
```

The following table describes the significant fields shown in the display.

Table 20: debug mpls ldp transport connections Field Descriptions

Field	Description
ldp:	Identifies the source of the message as LDP.
adj 0xnnnnnnnn	Identifies the data structure used to represent the peer at the transport level. Useful for correlating debug output.
a.b.c.d -> p.q.r.s	Indicates a TCP connection between a.b.c.d and p.q.r.s.
a.b.c.d:x -> p.q.r.s:y	Indicates a TCP connection between a.b.c.d, port x and p.q.r.s, port y.

Related Commands

Command	Description
debug mpls ldp transport events	Prints information about the events related to the LDP peer discovery mechanism.

debug mpls ldp transport events

To display information about events related to the label distribution protocol (LDP) peer discovery mechanism, use the **debugmplsldptransportevents**command in privilegedEXEC mode. This mechanism is used to determine the devices with which you wish to establish LDP sessions. To disable this feature, use the no form of this command.

debug mpls ldp transport events [**peer-acl** *acl*] [**interface**]
no debug mpls ldp transport events [**peer-acl** *acl*] [**interface**]

Syntax Description

peer-acl *acl*	(Optional) Limits the displayed information to that for LDP peers permitted by the access control list (*acl*).
interface	(Optional) Limits the displayed information to that for the specified interface.

Command Default

Displays information about LDP discovery activity for all peers and all interfaces.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.0(10)ST	This command was modified to reflect MPLS IETF command syntax and terminology.
12.0(14)ST	This command was integrated into Cisco IOS Release 12.0(14)ST.
12.1(2)T	This command was integrated into Cisco IOS Release 12.1(2)T.
12.1(8a)E	This command was integrated into Cisco IOS Release 12.1(8a)E.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(4)T	This command was integrated into Cisco IOS Release 12.2(4)T.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to monitor LDP discovery activity.

This command might generate a great deal of output. Use the peer-acl option or interface option, or both, to limit the output to peers or interfaces of interest.

Note The command includes all of the output generated by the debug mpls ldp transport connection command.

Examples

The following shows sample output from the**debugmplsldptransportevents** command:

```
Router#
debug mpls ldp transport events
ldp: enabling ldp on Ethernet1/1/1
ldp: Set intf id: intf 0x611D684C, Ethernet1/1/1, not lc-atm, intf_id 0
ldp: Set intf id: intf 0x617C5638, ATM0/0.2, not lc-atm, intf_id 0
ldp: Send ldp hello; ATM3/0.1, src/dst 8.1.1.1/224.0.0.2, inst_id 1, tcatm
ldp: Rcvd ldp hello; ATM3/0.1, from 203.0.7.7 (203.0.7.7:2), intf_id 1, opt 0x8, tcatm
ldp: Send ldp hello; Ethernet1/1/1, src/dst 138.1.0.88/224.0.0.2, inst_id 0
ldp: Rcvd ldp hello; Ethernet1/1/1, from 10.105.0.9 (7.1.1.1:0), intf_id 0, opt 0xC
ldp: ldp Hello from 10.105.0.9 (7.1.1.1:0) to 224.0.0.2, opt 0xC
ldp: New adj 0x617C5EBC from 10.105.0.9 (7.1.1.1:0), Ethernet1/1/1
ldp: Opening ldp conn; adj 0x617C5EBC, 8.1.1.1 <-> 7.1.1.1
ldp: ldp conn is up; adj 0x617C5EBC, 8.1.1.1:11013 <-> 7.1.1.1:646
ldp: Send ldp hello; ATM3/0.1, src/dst 8.1.1.1/224.0.0.2, inst_id 1, tcatm
ldp: Rcvd ldp hello; ATM3/0.1, from 203.0.7.7 (203.0.7.7:2), intf_id 1, opt 0x8, tcatm
ldp: Send ldp hello; Ethernet1/1/1, src/dst 138.1.0.88/224.0.0.2, inst_id 0
ldp: Rcvd ldp hello; Ethernet1/1/1, from 10.105.0.9 (7.1.1.1:0), intf_id 0, opt 0xC
...
ldp: Send ldp hello; Ethernet1/1/1, src/dst 138.1.0.88/224.0.0.2, inst_id 0
ldp: Send ldp hello; ATM3/0.1, src/dst 8.1no tag ip
.0.2, inst_id 1, tcatm
ldp: disabling ldp on Ethernet1/1/1
ldp: Hold timer expired for adj 0x617C5EBC, will close conn
ldp: Closing ldp conn 8.1.1.1:11013 <-> 7.1.1.1:646, adj 0x617C5EBC
ldp: Adjacency 0x617C5EBC, 10.105.0.9 timed out
ldp: Adj 0x617C5EBC; state set to closed
ldp: Rcvd ldp hello; ATM3/0.1, from 203.0.7.7 (203.0.7.7:2), intf_id 1, opt 0x8, tcatm
ldp: Ignore Hello from 10.105.0.9, Ethernet1/1/1; no intf
```

The following table describes the significant fields shown in the display.

Table 21: debug mpls ldp transport events Field Descriptions

Field	Description
ldp:	Identifies the source of the message as LDP.
adj 0xnnnnnnnn	Identifies the data structure used to represent the peer at the transport level. Useful for correlating debug output.
a.b.c.d (p.q.r.s:n)	Network address and LDP identifier of the peer.
intf_id	Interface identifier (non-zero for LC-ATM interfaces; 0 otherwise).

Field	Description
opt 0xn	Bits that describe options in the LDP discovery Hello packet: • 0x1--Targeted Hello option • 0x2--Send targeted Hello option • 0x4--Transport address option • 0x8--LDP Hello message (as opposed to TDP Hello message)

Related Commands

Command	Description
debug mpls ldp transport connections	Displays information about the TCP connections used to support LDP sessions.
show mpls ldp discovery	Displays the status of the LDP discovery process.

debug mpls lfib cef

To print detailed information about label rewrites being created, resolved, and deactivated as Cisco Express Forwarding (CEF) routes are added, changed, or removed, use the **debugmplslfibcef** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lfib cef
no debug mpls lfib cef

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI syntax.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Several lines of output are produced for each route placed into the label-forwarding information base (LFIB). If your router has thousands of labeled routes, be careful about issuing this command. When label switching is first enabled, each of these routes is placed into the LFIB, and several lines of output are displayed for each route.

Examples

The following is sample output from the **debugmplslfibcef**command**:**

```
Router# debug mpls lfib cef
Cisco Express Forwarding related TFIB services debugging is on
tagcon: tc_ip_rtlookup fail on 10.0.0.0/8:subnet_lookup failed
TFIB: route tag chg 10.7.0.7/32,idx=1,inc=Withdrn,outg=Withdrn,enabled=0x2
TFIB: fib complete delete: prefix=10.7.0.7/32,inc tag=26,delete_info=1
TFIB: deactivate tag rew for 10.7.0.7/32,index=0
TFIB: set fib rew: pfx 10.7.0.7/32,index=0,add=0,tag_rew->adj=Ethernet2/3
TFIB: resolve tag rew,prefix=10.7.0.7/32,no tag_info,no parent
TFIB: fib scanner start:needed:1,unres:0,mac:0,loadinfo:0
TFIB: resolve tag rew,prefix=10.7.0.7/32,no tag_info,no parent
TFIB: fib upd loadinf 10.100.100.100/32,tag=Tun_hd,fib no loadin,tfib no loadin
TFIB: fib check cleanup for 10.100.100.100/32,index=0,return_value=0
TFIB: fib_scanner_end
TFIB: create dynamic entry for 10.11.0.11/32
TFIB: call find_route_tags,dist_method=1,next_hop=10.93.0.11,Et2/3
TFIB: route tag chg 10.11.0.11/32,idx=0,inc=26,outg=Unkn,enabled=0x3
TFIB: create tag info 10.11.0.11/32,inc tag=26,has no info
TFIB: resolve tag rew,prefix=10.11.0.11/32,has tag_info,no parent
TFIB: finish fib res 10.11.0.11/32:index 0,parent outg tag no parent
TFIB: fib upd loadinf 10.11.0.11/32,tag=26,fib no loadin,tfib no loadin
TFIB: set fib rew: pfx 10.11.0.11/32,index=0,add=1,tag_rew->adj=Ethernet2/3
tagcon: route_tag_change for: 10.250.0.97/32
        intag 33, outtag 28, nexthop tsr 10.11.0.11:0
```

```
TFIB: route tag chg 10.250.0.97/32,idx=0,inc=33,outg=28,enabled=0x3
TFIB: deactivate tag rew for 10.250.0.97/32,index=0
TFIB: set fib rew: pfx 10.250.0.97/32,index=0,add=0,tag_rew->adj=Ethernet2/3
TFIB: create tag info 10.250.0.97/32,inc tag=33,has old info
On VIP:
TFIB: route tag chg 10.13.72.13/32,idx=0,inc=34,outg=Withdrn,enabled=0x3
TFIB: deactivate tag rew for 10.13.72.13/32,index=0
TFIB: set fib rew: pfx 10.13.72.13/32,index=0,add=0,tag_rew->adj=
TFIB: create tag info 10.13.72.13/32,inc tag=34,has old info
TFIB: resolve tag rew,prefix=10.13.72.13/32,has tag_info,no parent
TFIB: finish fib res 10.13.72.13/32:index 0,parent outg tag no parent
TFIB: set fib rew: pfx 10.100.100.100/32,index=0,add=0,tag_rew->adj=
TFIB: create tag info 10.100.100.100/32,inc tag=37,has old info
TFIB: resolve tag rew,prefix=10.100.100.100/32,has tag_info,no parent
TFIB: finish fib res 10.100.100.100/32:index 0,parent outg tag no parent
TFIB: fib upd loadinf 10.100.100.100/32,tag=37,fib no loadin,tfib no loadin
```

The following table lists the significant fields and a description of special labels that appear in the output of this **debug** command shown in the display.

Table 22: debug mpls lfib cef Field Descriptions

Field	Description
tagcon	The name of the subsystem issuing the debug output (Label Control).
LFIB	The name of the subsystem issuing the debug output.
tc_ip_rtlookup fail on x.y.w.z/m: subnet_lookup failed	The destination with IP address and mask shown is not in the routing table.
route tag chg x.y.w.z/m	Request to create the LFIB entry for the specified prefix/mask.
idx=-1	The index within the FIB entry of the path whose LFIB entry is being created. The parameter -1 means all paths for this FIB entry.
inc=s	Incoming label of the entry being processed.
outg=s	Outgoing label of the entry being processed.
enabled=0xn	Bit mask indicating the types of label switching currently enabled: • 0x1 = dynamic • 0x2 = TSP tunnels • 0x3 = both
fib complete delete	Indicates that the FIB entry is being deleted.
prefix=x.y.w.z/m	A destination prefix.
delete_info=1	Indicates that label_info is also being deleted.
deactivate tag rew for x.y.w.z/m	Indicates that label rewrite for specified prefix is being deleted.
index=n	Index of path in the FIB entry being processed.

Field	Description
set fib rew: pfx x.y.w.z/m	Indicates that label rewrite is being installed or deleted from the FIB entry for the specified destination for label imposition purposes.
add=0	Indicates that label rewrite is being deleted from the FIB (no longer imposing labels).
tag_rew->adj=s	Adjacency of label rewrite for label imposition.
resolve tag rew,prefix=x.y.w.z/m	Indicates that the FIB route to the specified prefix is being resolved.
no tag_info	Indicates that there is no label_info for the destination (destination not labeled).
no parent	Indicates that the route is not recursive.
fib scanner start	Indicates that the periodic scan of the FIB has started.
needed:1	Indicates that the LFIB needs the FIB to be scanned.
unres:n	Indicates the number of unresolved TFIB entries.
mac:n	Indicates the number of TFIB entries missing MAC strings.
loadinfo:n	Indicates whether the nonrecursive accounting state has changed and whether the loadinfo information in the LFIB needs to be adjusted.
fib upd loadinf x.y.w.z/m	Indicates that a check for nonrecursive accounting is being made and that the LFIB loadinfo information for the specified prefix is being updated.
tag=s	Incoming label of entry.
fib no loadin	Indicates that the corresponding FIB entry has no loadinfo.
tfib no loadin	Indicates that the LFIB entry has no loadinfo.
fib check cleanup for x.y.w.z/m	Indicates that a check is being made on the LFIB entry for the specified destination to determine if rewrite needs to be removed from the LFIB.
return_value=x	If x is 0, indicates that no change has occurred in the LFIB entry. If x is 1, there was a change.
fib_scanner_end	Indicates that the FIB scan has come to an end.
create dynamic entry for x.y.w.z/m	Indicates that the LFIB has been enabled and that an LFIB entry is being created for the specified destination.
call find_route_tags	Indicates that the labels for that destination are being requested.
dist_method=n	Identifies the label distribution method--TDP, TC-ATM, and so on.
next_hop=x.y.z.w	Identifies the next hop for the destination.
interface name	Identifies the outgoing interface for the destination.

Field	Description
create tag info	Indicates that a label_info data structure is being created for the destination.
has no info	Indicates that the destination does not already have label_info.
finish fib re x.y.z.w/m	Indicates that the LFIB entry for the specified route is being completed.
parent outg tag s	If recursive, specifies the outgoing label of the route through which it is recursive (the parent). If not recursive, s = "no parent."
tagcon: route_tag_change for: x.y.z.w/m	Indicates that label control is notifying LFIB that labels are available for the specified destination.
intag s	Identifies the incoming label for the destination.
outtag s	Identifies the outgoing label for the destination.
nexthop tsr x.y.z.w.i	Identifies the TDP ID of the next hop that sent the tag.

Related Commands

Command	Description
debug mpls lfib lsp	Prints detailed information about label rewrites being created and deleted as LSP tunnels are added or removed.
debug mpls lfib state	Traces what happens when label switching is enabled or disabled.
debug mpls lfib struct	Traces the allocation and freeing of LFIB-related data structures, including the LFIB itself, label rewrites, and label_info data.

debug mpls lfib enc

To print detailed information about label encapsulations while label rewrites are created or updated and placed in the label-forwarding information base (LFIB), use the **debugmplslfibenc** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lfib enc
no debug mpls lfib enc

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI syntax.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Several lines of output are produced for each route placed into the LFIB. If your router has thousands of labeled routes, issue this command with care. When label switching is first enabled, each of these routes is placed into the LFIB and a label encapsulation is created. The command output shows you on which adjacency the label rewrite is being created and the labels assigned.

Examples

The following is sample output from the **debugmplslfibenc** command. This example shows the encapsulations for three routes that have been created and placed into the LFIB.

```
Router# debug mpls lfib enc
TFIB: finish res:inc tag=28,outg=Imp_null,next_hop=10.93.72.13,Ethernet4/0/3
TFIB: update_mac, mac_length = 14,addr=10.93.72.13,idb=Ethernet4/0/3
TFIB: get ip adj: addr=10.93.72.13,is_p2p=0,fibidb=Ethernet4/0/3,linktype=7
TFIB: get tag adj: addr=10.93.72.13,is_p2p=0,fibidb=Ethernet4/0/3,linktype=79
TFIB: encaps:inc=28,outg=Imp_null,idb:Ethernet4/0/3,sizes 14,14,1504,type 0
TFIB: finish res:inc tag=30,outg=27,next_hop=10.93.72.13,Ethernet4/0/3
TFIB: get ip adj: addr=10.93.72.13,is_p2p=0,fibidb=Ethernet4/0/3,linktype=7
TFIB: get tag adj: addr=10.93.72.13,is_p2p=0,fibidb=Ethernet4/0/3,linktype=79
TFIB: encaps:inc=30,outg=27,idb:Ethernet4/0/3,sizes 14,18,1500,type 0
TFIB: finish res:inc tag=30,outg=10,next_hop=0.0.0.0,ATM0/0.1
TFIB: get ip adj: addr=0.0.0.0,is_p2p=1,fibidb=ATM0/0.1,linktype=7
TFIB: get tag adj: addr=0.0.0.0,is_p2p=1,fibidb=ATM0/0.1,linktype=79
TFIB: encaps:inc=30,outg=10,idb:ATM0/0,sizes 4,8,4470,type 1
```

The following table describes the significant fields shown in the display.

Table 23: debug mpls lfib enc Field Descriptions

Field	Description
TFIB	Identifies the source of the message as the LFIB subsystem.
finish res	Identifies that the LFIB resolution is being finished.
inc tag=x or inc=x	An incoming (local) label for the LFIB entry is being created. Labels can be numbers or special values.
outg=y	An outgoing (remote) label for the LFIB entry is being created.
next_hop=a.b.c.d	IP address of the next hop for the destination.
interface	The outgoing interface through which a packet will be sent.
get ip adj	Identifies that the IP adjacency to use in the LFIB entry is being determined.
get tag adj	Identifies that the label switching adjacency to use for the LFIB entry is being determined.
addr = a.b.c.d	The IP address of the adjacency.
is_p2p=x	If x is 1, this is a point-to-point adjacency. If x is 0, it is not.
fibidb = s	Indicates the interface of the adjacency.
linktype = x	The link type of the adjacency, as follows: • 7 = LINK_IP • 79 = LINK_TAG
sizes x,y,z	Indicates the following values: • x = length of macstring • y = length of tag encapsulation • z = tag MTU
type = x	Tag encapsulation type, as follows: • 0 = normal • 1 = TCATM • 2 = TSP tunnel
idb:s	Indicates the outgoing interface.
update_mac	Indicates that the macstring of the adjacency is being updated.

The following table describes the special labels, which sometimes appear in the debug output, and their meanings.

Table 24: Special Labels Appearing in debug Command Output

Special Label	Meaning
Unassn--Inital value	No label assigned yet.
Unused	This destination does not have a label (for example, a BGP route).
Withdrn	The label for this destination has been withdrawn.
Unkn	This destination should have a label, but it is not yet known.
Get_res	A recursive route that will get a label when resolved.
Exp_null	Explicit null label--used over TC-ATM.
Imp_null	Implicit null label--for directly connected routes.
Tun_hd	Identifies head of TSP tunnel.

Related Commands

Command	Description
debug mpls lfib cef	Prints detailed information about label rewrites being created, resolved, and deactivated as CEF routes are added, changed, or removed.
debug mpls lfib lsp	Prints detailed information about label rewrites being created and deleted as LSP tunnels are added or removed.
debug mpls lfib state	Traces what happens when label switching is enabled or disabled.
debug mpls lfib struct	Traces the allocation and freeing of LFIB-related data structures, including the LFIB itself, label rewrites, and label_info data.

debug mpls lfib fast-reroute database

To enable debugging information about changes to the fast reroute database, use the debug mpls lfib fast-reroute database command in privileged EXEC command. To disable debugging output, use to no form of this command.

debug mpls lfib fast-reroute database
no debug mpls lfib fast-reroute database

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(10)ST	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Because using debug commands can use a great deal of bandwidth on your system, use caution when enabling the debug mpls lfib fast-reroute database command.

Examples

The following example displays debugging output for in Fast Reroute processing:

```
Router# debug mpls lfib fast-reroute database
LFIB-FRR:Clear headend FRR info for Tunnel1
LFIB-FRR:FRR info for Tunnel1 changed
LFIB-FRR:update headend FRR info for 10.8.0.1/32
LFIB-FRR:item B13D94 [Tu1] (group PO0/0->Tu4000):destroying entry for 10.8.0.1/32... [514
left]
LFIB-FRR:item B13D94 [Tu1]:removed from name tree
LFIB-FRR:item B13D94 [Tu1]:removed from group PO0/0->Tu4000 tree
%LINK-5-CHANGED:Interface Tunnel1, changed state to administratively down
LFIB-FRR:Clear headend FRR info for Tunnel1
LFIB-FRR:FRR info for Tunnel1 changed
LFIB-FRR:Clear headend FRR info for Tunnel1
LFIB-FRR:FRR info for Tunnel1 changed
LFIB-FRR:Set headend FRR info for Tunnel1 {main=PO0/0,backup=Tu4000,label=18}
LFIB-FRR:FRR info for Tunnel1 changed
%SYS-5-CONFIG_I:Configured from console by console
LFIB-FRR:update headend FRR info for 10.8.0.1/32
LFIB-FRR:item B13D94 [Tu1]:inserted in name tree
LFIB-FRR:item B13D94 [Tu1]:inserted in group PO0/0->Tu4000 tree
LFIB-FRR:item B13D94 [Tu1] (group PO0/0->Tu4000):full entry created for 10.8.0.1/32 [total
 515]
LFIB-FRR:update headend FRR info for 10.8.0.1/32
LFIB-FRR:item B13D94 [Tu1] (group PO0/0->Tu4000):updating entry for 10.8.0.1/32...
LFIB-FRR:item B13D94 [Tu1] (group PO0/0->Tu4000):... updated
%LINK-3-UPDOWN:Interface Tunnel1, changed state to up
LFIB-FRR:update headend FRR info for 10.43.0.0/16
LFIB-FRR:item B04C2C [Tu486]:inserted in name tree
LFIB-FRR:item B04C2C [Tu486]:inserted in group PO0/0->Tu4000 tree
```

```
LFIB-FRR:item B04C2C [Tu486] (group PO0/0->Tu4000):full entry created for 10.43.0.0/16
[total 516]
LFIB-FRR:update headend FRR info for 10.43.0.0/16
LFIB-FRR:item B04BB4 [Tu481]:inserted in name tree
LFIB-FRR:item B04BB4 [Tu481]:inserted in group PO0/0->Tu4000 tree
LFIB-FRR:item B04BB4 [Tu481] (group PO0/0->Tu4000):full entry created for 10.43.0.0/16
[total 517]
LFIB-FRR:update headend FRR info for 10.2.0.0/16
LFIB-FRR:item B04B3C [Tu486]:inserted in name tree
LFIB-FRR:item B04B3C [Tu486]:inserted in group PO0/0->Tu4000 tree
LFIB-FRR:item B04B3C [Tu486] (group PO0/0->Tu4000):full entry created for 10.2.0.0/16 [total
 518]
LFIB-FRR:update headend FRR info for 10.2.0.0/16
LFIB-FRR:item B04AC4 [Tu481]:inserted in name tree
LFIB-FRR:item B04AC4 [Tu481]:inserted in group PO0/0->Tu4000 tree
```

Related Commands

Command	Description
debug mpls traffic-eng tunnels fast-reroute events	Displays debugging information about fast reroute events.
debug mpls traffic-eng tunnels fast-reroute reroutes	Displays debugging information about the rerouting of traffic from link-protected interfaces to backup tunnels.

debug mpls lfib fast-reroute events

To display debugging information about fast reroute events, use the debug mpls lfib fast-reroute events command in privileged EXEC command. To disable debugging output, use to no form of this command.

debug mpls lfib fast-reroute events
no debug mpls lfib fast-reroute events

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(10)ST	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Because using debug commands can use a great deal of bandwidth on your system, use caution when enabling the debug mpls lfib fast-reroute events command.

Examples

The following example reports on the rerouting of traffic to a backup tunnel because of a change of state at a link-protected physical interface.

```
Router# debug mpls lfib fast-reroute events
LFIB-FRR:enqueued interface DOWN event for PO0/0 (Up)
LFIB-FRR:discarded interface DOWN event for PO0/0 (Up)
LFIB-FRR:processing interface DOWN event for PO0/0 (Up)
LFIB-FRR:group PO0/0->Tu4000:output if fixup:Backup(Tu4000) -> Backup(Tu4000)
```

Note The state given in parentheses reflects what the FRR database currently understands to be the state of the physical interface. This may or may not be the same as the event state reported earlier on that same display line.

Related Commands

Command	Description
debug mpls traffic-eng tunnels fast-reroute database	Displays debugging information about changes to the fast reroute database.
debug mpls traffic-eng tunnels fast-reroute reroutes	Displays debugging information about the rerouting of traffic from link-protected interfaces to backup tunnels.

debug mpls lfib fast-reroute reroutes

To enable debugging information about the rerouting of protected Label Forwarding Information Base (LFIB) entries between the primary and backup outgoing interfaces, use the debug mpls lfib fast-reroute reroutes command in privileged EXEC command. To disable debugging output, use to no form of this command.

debug mpls lfib fast-reroute reroutes
no debug mpls lfib fast-reroute reroutes

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(10)ST	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Because using debug commands can use a great deal of bandwidth on your system, use caution when enabling the debug mpls lfib fast-reroute reroutes command. The output of this command increases in proportion to the number of tunnels that utilize fast reroute.

Examples

The following example reports the results of reroute attempts:

```
Router# debug mpls lfib fast-reroute reroutes
LFIB-FRR:item B0E844 [Tu139]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0E8BC [Tu138]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0E934 [Tu387]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0E9AC [Tu137]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0EA24 [Tu136]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0EA9C [Tu135]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0EB14 [Tu384]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0EB8C [Tu134]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0EC04 [Tu133]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
LFIB-FRR:item B0EC7C [Tu132]:output if fixup, Main(PO0/0) -> Backup(Tu4000), succeeded
```

Related Commands

Command	Description
debug mpls traffic-eng tunnels fast-reroute database	Displays debugging information about changes to the fast reroute database.
debug mpls traffic-eng tunnels fast-reroute events	Displays debugging information about fast reroute events.

debug mpls lfib lsp

To print detailed information about label rewrites being created and deleted as label-switched path (LSP) tunnels are added or removed, use the **debugmplslfiblsp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lfib lsp
no debug mpls lfib lsp

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI syntax.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debugmplslfiblsp** command:

```
Router# debug mpls lfib lsp
TSP-tunnel related TFIB services debugging is on
TFIB: tagtun,next hop=10.93.72.13,inc=35,outg=1,idb=Et4/0/3
TFIB: tsptunnel:next hop=10.93.72.13,inc=35,outg=Imp_null,if_number=7
TFIB: tsptun update loadinfo:tag=35,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun tag chg linec,fiblc=0,in tg=35,o tg=1,if=7,nh=10.93.72.13
TFIB: tagtun,next hop=10.92.0.7,inc=36,outg=1,idb=Et4/0/2
TFIB: tsptunnel:next hop=10.92.0.7,inc=36,outg=Imp_null,if_number=6
TFIB: tsptun update loadinfo:tag=36,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun tag chg linec,fiblc=0,in tg=36,o tg=1,if=6,nh=10.92.0.7
TFIB: tagtun_delete, inc = 36
tagtun tag del linec,itag=12
TFIB: tagtun_delete, inc = 35
tagtun tag del linec,itag=12
TFIB: tagtun,next hop=10.92.0.7,inc=35,outg=1,idb=Et4/0/2
TFIB: tsptunnel:next hop=10.92.0.7,inc=35,outg=Imp_null,if_number=6
TFIB: tsptun update loadinfo:tag=35,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun tag chg linec,fiblc=0,in tg=35,o tg=1,if=6,nh=10.92.0.7
On VIP:
TFIB: tagtun chg msg,in tg=35,o tg=1,nh=10.93.72.13,if=7
TFIB: tsptunnel:next hop=10.93.72.13,inc=35,outg=Imp_null,if_number=7
TFIB: tsptun update loadinfo:tag=35,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun chg msg,in tg=36,o tg=1,nh=10.92.0.7,if=6
TFIB: tsptunnel:next hop=10.92.0.7,inc=36,outg=Imp_null,if_number=6
TFIB: tsptun update loadinfo:tag=36,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun chg msg,in tg=35,o tg=1,nh=10.93.72.13,if=7
TFIB: tsptunnel:next hop=10.93.72.13,inc=35,outg=Imp_null,if_number=7
TFIB: tsptun update loadinfo:tag=35,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun chg msg,in tg=36,o tg=1,nh=10.92.0.7,if=6
TFIB: tsptunnel:next hop=10.92.0.7,inc=36,outg=Imp_null,if_number=6
```

```
TFIB: tsptun update loadinfo:tag=36,loadinfo_reqd=0,no new loadinfo,no old loadinfo
TFIB: tagtun chg msg,in tg=35,o tg=1,nh=10.92.0.7,if=6
TFIB: tsptunnel:next hop=10.92.0.7,inc=35,outg=Imp_null,if_number=6
TFIB: tsptun update loadinfo:tag=35,loadinfo_reqd=0,no new loadinfo,no old loadinfo
```

The following table describes the significant fields shown in the sample display.

Table 25: debug mpls lfib lsp Field Descriptions

Field	Description
tagtun	Name of routine entered.
next hop=x.y.z.w	Next hop for the tunnel being created.
inc=x	Incoming label for this hop of the tunnel being created.
outg=x	Outgoing label (1 means Implicit Null label).
idb=s	Outgoing interface for the tunnel being created.
if_number=7	Interface number of the outgoing interface.
tsptunnel	Name of the routine entered.
tsptun update loadinfo	The procedure being performed.
tag=x	Incoming label of the LFIB slot whose loadinfo is being updated.
loadinfo_reqd=x	Indicates whether a loadinfo is expected for this entry (non-recursive accounting is on).
no new loadinfo	No change required in loadinfo.
no old loadinfo	No previous loadinfo available.
tagtun tag chg linec	Line card is being informed of the TSP tunnel.
fiblc=x	Indicates which line card is being informed (0 means all).
in tg=x	Indicates the incoming label of new TSP tunnel.
o tg=x	Indicates the outgoing label of new TSP tunnel.
if=x	Indicates the outgoing interface number.
nh=x.y.w.z	Indicates the next hop IP address.
tagtun_delete	Indicates that a procedure is being performed: delete a TSP tunnel.
tagtun tag del linec	Informs the line card of the TSP tunnel deletion.
tagtun chg msg	Indicates that the line card has received a message to create a TSP tunnel.

Related Commands

Command	Description
debug mpls lfib cef	Prints detailed information about label rewrites being created, resolved, and deactivated as CEF routes are added, changed, or removed.
debug mpls lfib state	Traces what happens when label switching is enabled or disabled.
debug mpls lfib struct	Traces the allocation and freeing of LFIB-related data structures, including the LFIB itself, label rewrites, and label_info data.

debug mpls lfib state

To trace what happens when label switching is enabled or disabled, use the **debugmplslfibstate**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lfib state
no debug mpls lfib state

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI syntax.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command when you wish to trace what happens to the label-forwarding information base (LFIB) when you issue the **mplsip** or the **mplstsp-tunnel** command.

Examples

The following is sample output from the **debugmplslfibstate** command:

```
Router# debug mpls lfib state
TFIB enable/disable state debugging is on
TFIB: Upd tag sb 6(status:0xC1,tmtu:1500,VPI:1-1 VC=0/32,et:0/0/0),lc 0x0
TFIB: intf status chg: idb=Et4/0/2,status=0xC1,oldstatus=0xC3
TFIB: interface dyntag change,change in state to Ethernet4/0/2
TFIB: enable entered, table exists,enabler type=0x2
TFIB: enable, TFIB already enabled, types now 0x3,returning
TFIB: enable entered, table exists,enabler type=0x1
TFIB: disable entered, table exists,type=0x1
TFIB: cleanup: tfib[32] still non-0
On linecard only:
TFIB: disable lc msg recvd, type=0x1
TFIB: Ethernet4/0/1 fibidb subblock message received
TFIB: enable lc msg recvd, type=0x1
TFIB: Tunnel301 set encapfix to 0x6016A97C
```

The following table describes the significant fields shown in the display.

Table 26: debug mpls lfib state Field Descriptions

Field	Description
LFIB	Identifies the source of the message as the LFIB subsystem.

Field	Description
Upd tag sb x	Indicates that the status of the "xth" label switching sub-block is being updated, where x is the interface number. There is a label switching sub-block for each interface on which label switching has been enabled.
(status:0xC1,tmtu:1500,VPI:1-1VC=0/32, et:0/0/0),lc 0x0)	Identifies the values of the fields in the label switching sub-block, as follows: • status byte • maximum transmission unit (*tmtu*) • range of ATM VPs • control VP • control VC (if this is a TC-ATM interface) • encapsulation type (*et*) • encapsulation information • tunnel interface number (*lc*) • line card number to which the update message is being sent (0 means all line cards)
intf status chg	Indicates that there was an interface status change.
idb=Et4/0/2	Identifies the interface whose status changed.
status=0xC1	Indicates the new status bits in the label switching sub-block of the idb.
oldstatus=0xC3	Indicates the old status bits before the change.
interface dyntag change, change in state to Ethernet4/0/2	Indicates that there was a change in the dynamic label status for the particular interface.
enable entered	Indicates that the code that enables the LFIB was invoked.
TFIB already enabled	Indicates that the LFIB was already enabled when this call was made.
table exists	Indicates that an LFIB table had already been allocated in a previous call.
cleanup: tfib[x] still non-0	Indicates that the LFIB is being deleted, but that slot x is still active.
disable lc mesg recvd, type=0x1	Indicates that a message to disable label switching type 1 (dynamic) was received by the line card.
disable entered, table exists,type=0x1	Indicates that a call to disable dynamic label switching was issued.

Field	Description
Ethernet4/0/1 fibidb subblock message received	Indicates that a message giving fibidb status change was received on the line card.
enable lc msg recvd,type=0x1	Indicates that the line card received a message to enable label switching type 1 (dynamic).
Tunnel301 set encapfix to 0x6016A97C	Shows that fibidb Tunnel301 on the line card received an encapsulation fixup.
types now 0x3, returning	Shows the value of the bitmask indicating the type of label switching enabled on the interface, as follows: • 0x1--means dynamic label switching • 0x2--means tsp-tunnels • 0x3--means both

Related Commands

Command	Description
debug mpls lfib cef	Prints detailed information about label rewrites being created, resolved, and deactivated as CEF routes are added, changed, or removed.
debug mpls lfib lsp	Prints detailed information about label rewrites being created and deleted as LSP tunnels are added or removed.
debug mpls lfib struct	Traces the allocation and freeing of LFIB-related data structures, including the LFIB itself, label rewrites, and label_info data.

debug mpls lfib struct

To trace the allocation and freeing of label-forwarding information base (LFIB)-related data structures, such as the LFIB itself, label rewrites, and label_info data, use the **debugmplslfibstruct** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lfib struct
no debug mpls lfib struct

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS IETF terminology and CLI syntax.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debugmplslfibstruct** command:

```
Router# debug mpls lfib struct
TFIB data structure changes debugging is on
TFIB: delete tag rew, incoming tag 32
TFIB: remove from tfib,inc tag=32
TFIB: set loadinfo,tag=32,no old loadinfo,no new loadinfo
TFIB: TFIB not in use.  Checking for entries.
TFIB: cleanup: tfib[0] still non-0
TFIB: remove from tfib,inc tag=Tun_hd
TFIB: set loadinfo,tag=Exp_null,no old loadinfo,no new loadinfo
TFIB: TFIB freed.
TFIB: enable, TFIB allocated, size 4024 bytes, maxtag = 500
TFIB: create tag rewrite: inc Tun_hd,outg Unkn
TFIB: add to tfib at Tun_hd, first in circular list, mac=0,enc=0
TFIB: delete tag rew, incoming tag Tun_hd
TFIB: remove from tfib,inc tag=Tun_hd
TFIB: set loadinfo,tag=Exp_null,no old loadinfo,no new loadinfo
TFIB: create tag rewrite: inc Tun_hd,outg Unkn
TFIB: add to tfib at Tun_hd, first in circular list, mac=0,enc=0
TFIB: create tag rewrite: inc 26,outg Unkn
TFIB: add to tfib at 26, first in circular list, mac=0,enc=0
TFIB: add to tfib at 27, added to circular list, mac=0,enc=0
TFIB: delete tag rew, incoming tag Tun_hd
TFIB: remove from tfib,inc tag=Tun_hd
TFIB: set loadinfo,tag=Exp_null,no old loadinfo,no new loadinfo
TFIB: add to tfib at 29, added to circular list, mac=4,enc=8
TFIB: delete tag rew, incoming tag 29
TFIB: remove from tfib,inc tag=29
```

The following table describes the significant fields shown in the display.

Table 27: debug mpls lfib struct Field Descriptions

Field	Description
TFIB	The subsystem issuing the message.
delete tag rew	A label rewrite is being freed.
remove from tfib	A label rewrite is being removed from the LFIB.
inc tag=s	The incoming label of the entry being processed.
set loadinfo	The loadinfo field in the LFIB entry is being set (used for nonrecursive accounting).
tag=s	The incoming label of the entry being processed.
no old loadinfo	The LFIB entry did not have a loadinfo before.
no new loadinfo	The LFIB entry should not have a loadinfo now.
TFIB not in use. Checking for entries.	Label switching has been disabled and the LFIB is being freed up.
cleanup: tfib[x] still non-0	The LFIB is being checked for any entries in use, and entry x is the lowest numbered slot still in use.
TFIB freed	The LFIB table has been freed.
enable, TFIB allocated, size x bytes, maxtag = y	Label switching has been enabled and an LFIB of x bytes has been allocated. The largest legal label is y.
create tag rewrite	A label rewrite is being created.
inc s	The incoming label.
outg s	The outgoing label.
add to tfib at s	A label rewrite has been placed in the LFIB at slots.
first in circular list	This LFIB slot had been empty and this is the first rewrite in the list.
mac=0,enc=0	Length of the MAC string and total encapsulation length, including labels.
added to circular list	A label rewrite is being added to an LFIB slot that already had an entry. This rewrite is being inserted in the circular list.

Related Commands

Command	Description
debug mpls lfib cef	Prints detailed information about label rewrites being created, resolved, and deactivated as CEF routes are added, changed, or removed.
debug mpls lfib lsp	Prints detailed information about label rewrites being created and deleted as LSP tunnels are added or removed.

Command	Description
debug mpls lfib state	Traces what happens when label switching is enabled or disabled.

debug mpls lspv

To display information related to the Multiprotocol Label Processing (MPLS) label switched path (LSP) Ping/Traceroute feature, use the **debugmplslspv**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls lspv [**tlv**] [**error**] [**event**] [**ipc**] [**packet** [{**data** | **error**}]] [**path-discovery**] [**multipath**] [**all**]
no debug mpls lspv

Syntax Description

tlv	(Optional) Displays MPLS) echo packet type, length, values (TLVs) information as it is being coded and decoded.
error	(Optional) Displays error conditions encountered during MPLS echo request and echo reply encoding and decoding. See the table below.
event	(Optional) Displays MPLS echo request and reply send and receive event information.
ipc	(Optional) Interprocess communication. Displays debug information regarding communication between the Route Processor and line cards.
packet data	(Optional) Displays detailed debugging information for the MPLS echo packets sent and received. This output is seen only on the originating router and the router generating the reply.
packet error	(Optional) Displays packet errors for MPLS echo request and reply. No output is expected for this command.
path-discovery	(Optional) Provides information regarding LSP traceroute path discovery operations.
multipath	(Optional) Displays multipath information.
all	(Optional) Enables all the command keywords.

Command Default

MPLS LSP debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(27)S	This command was introduced.
12.4(6)T	The following keywords were added: **ipc**, **path-discovery**, **multipath**, and **all**.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.28(SB) and implemented on the Cisco 10000 series router.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(31)SB2	This command was integrated into Cisco IOS Release 12.2(31)SB2.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.0(33)S	The following keywords were added for Cisco IOS Release 12.0(33)S: **ipc**, **path-discovery**, **multipath**, and **all**.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

Use this command to monitor activity associated with the **pingmpls** and the **tracempls** commands.

The following table lists the messages displayed by the **debugmplslspverror**command and the reason for each error message.

Table 28: Messages Displayed by the debug mpls lspv error Command

Message	Reason Why Message Is Displayed
Echo reply discarded because not routable	An echo reply message is sent because the IP header indicates that the packet has the Router Alert set and the packet is not routable.
UDP checksum error, packet discarded	A packet is received on the port being used by Label Switched Path Verification (LSPV) and there is a checksum error on the packet.
Invalid echo message type	An MPLS echo packet with an invalid echo message type (neither a request nor a reply) is received.
Illegal Action	The state machine that drives the LSPV software detects an invalid condition.

Examples

The following is sample output from the**pingmpls** command when LSPV event debugging is enabled:

```
Router# debug mpls lspv event
LSPV event debugging is on
Router# ping mpls ipv4 10.131.159.252/32 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 48/48/48 ms
Router#
*Dec 31 19:31:15.366: LSPV:
waiting for 2 seconds
*Dec 31 19:31:15.366: LSPV: sender_handle: 2000002D, Event Echo Requests Start,
[Idle->Waiting for Echo Reply]
*Dec 31 19:31:15.414: LSPV: sender_handle: 2000002D, Event Echo Reply Received,
```

```
[Waiting for Echo Reply->Waiting for Interval]
*Dec 31 19:31:15.466: LSPV: sender_handle: 2000002D, Event Echo Requests Cancel,
[Waiting for Interval->Idle]
Router# undebug all
All possible debugging has been turned off
```

The following is sample output from the**pingmpls** command when LSPV TLV debugging is enabled:

```
Router# debug mpls lspv tlv
LSPV tlv debugging is on
Router# ping mpls ipv4 10.131.159.252/32 repeat 1
Sending 1, 100-byte MPLS Echos to 10.131.159.252/32,
      timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target
Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 40/40/40 ms
Router#
*Dec 31 19:32:32.566: LSPV: Echo Hdr encode: version 1, msg type 1, reply mode 2
, return_code 0, return_subcode 0, sender handle 9400002E, sequence number 1,
timestamp sent 14:32:32 EST Wed Dec 31 2003, timestamp rcvd 19:00:00 EST Thu Dec 31 1899
*Dec 31 19:32:32.566: LSPV: IPV4 FEC encode: destaddr 10.131.159.252/32
*Dec 31 19:32:32.566: LSPV: Pad TLV encode: type 1, size 18, pattern 0xABCD
*Dec 31 19:32:32.606: LSPV: Echo Hdr decode: version 1, msg type 2, reply mode 2,
return_code 3, return_subcode 0, sender handle 9400002E, sequence number 1,
timestamp sent 14:32:32 EST Wed Dec 31 2003, timestamp rcvd 14:32:32 EST Wed Dec 31 2003
Router# undebug all
All possible debugging has been turned off
```

The following is sample output from the **tracemplsmultipath** command when LSPV multipath debugging is on:

```
Router# debug mpls lspv multipath
multipath information debugging is on
Router# trace mpls multipath ipv4 10.5.5.5/32

Starting LSP Multipath Traceroute for 10.5.5.5/32
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
LL
*Aug 30 20:39:03.719: LSPV: configuring bitmask multipath, base 0x7F000000, bitmapsize 32,
 start 0x7F000000, numbits 32
*Aug 30 20:39:03.719: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
 8, start 127.0.0.0, base 127.0.0.0, numbits 32
*Aug 30 20:39:03.719: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
 8, start 127.0.0.0, base 127.0.0.0, numbits 32
*Aug 30 20:39:03.719: LSPV: getnext bit_cursor 0, index 0, mask 0x80000000
*Aug 30 20:39:03.719: LSPV: next addr 127.0.0.1
*Aug 30 20:39:03.719: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:03.719: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.007: LSPV: multipath info: !
Path 0 found,
 output interface Et1/0 source 10.2.3.2 destination 127.0.0.1
Paths (found/broken/unexplored) (1/0/0)
 Echo Request (sent/fail) (3/0)
```

```
 Echo Reply (received/timeout) (3/0)
 Total Time Elapsed 924 ms
Router#
*Aug 30 20:39:04.007: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.007: LSPV: ds map convert: rtr_id A030404, mtu 1500 intf_addr 10.3.4.4
hashkey 8, multipath length 8, info 2130706432
*Aug 30 20:39:04.007: LSPV: multipath info: hashkey type 8, base 0x7F000000, bitmapsize 32,
 info0 0xFFFFFFFF
*Aug 30 20:39:04.007: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
 8, start 127.0.0.1, base 127.0.0.1, numbits 32
*Aug 30 20:39:04.007: LSPV: getnext bit_cursor 0, index 0, mask 0x80000000
*Aug 30 20:39:04.007: LSPV: next addr 127.0.0.1
*Aug 30 20:39:04.007: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:04.007: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.299: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:04.299: 7F 00 00 00 FF FF FF FF
*Aug 30 20:39:04.299: LSPV: ds map convert: rtr_id A040505, mtu 1504 intf_addr 10.4.5.5
hashkey 8, multipath length 8, info 2130706432
*Aug 30 20:39:04.299: LSPV: multipath info: hashkey type 8, base 0x7F000000, bitmapsize 32,
 info0 0xFFFFFFFF
*Aug 30 20:39:04.299: LSPV: multipath info: info_length 4, bitmapsize 32, multipath_length
 8, start 127.0.0.1, base 127.0.0.1, numbits 32
*Aug 30 20:39:04.299: LSPV: getnext bit_cursor 0, index 0, mask 0x80000000
*Aug 30 20:39:04.299: LSPV: next addr 127.0.0.1
*Aug 30 20:39:04.299: LSPV: multipath info: datagramsize 8
*Aug 30 20:39:04.299: 7F 00 00 00 FF FF FF FF
Router# undebug all

multipath information debugging is off
```

Related Commands

Command	Description
ping mpls	Checks MPLS LSP connectivity.
trace mpls	Discovers MPLS LSP routes that packets will actually take when traveling to their destinations.

debug mpls mldp all

To enable debugging output for all Multicast Label Distribution Protocol (MLDP) events, use the **debug mpls mldp all**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp all
no debug mpls mldp all

Syntax Description

This command has no arguments or keywords.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for all MLDP events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network MVPN core network. Issuing this command is equivalent to issuing the following commands:

- debug mpls mldp filter opaque_type
- debug mpls mldp generic
- debug mpls mldp gr
- debug mpls mldp mfi
- debug mpls mldp mrib
- debug mpls mldp neighbor
- debug mpls mldp packet

Examples

The following example shows how to enable debugging output for all MLDP events:

```
Router# debug mpls mldp all
```

Related Commands

Command	Description
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.

Command	Description
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp filter opaque_type

To enable filtering of Multicast Label Distribution Protocol (MLDP) debugging output using the opaque type, use the **debug mpls mldp filter opaque_type**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp filter opaque_type *type*
no debug mpls mldp opaque_type *type*

Syntax Description

type	The opaque type to be used for filtering. The following types are supported: • **ipv4** *source-group* --this represents the "IPv4 Protocol Independent Source-Specific Transit" multicast application type. The IPv4 source address and group address are also specified. • **ipv6** *source-group* --this represents the "IPv6 Protocol Independent Source-Specific Transit" multicast application type.The IPv6 source address and group address are also specified. • **mdt** *vpn-id mdt-number* --this represents the "Multicast Virtual Private Network (MVPN)" multicast application type. The VPN identifier and the Multicast Distribution Tree (MDT) number are also specified. • **vpnv4** *source-group route-distinguisher* --this represents the "Direct MDT (VPNv4) " multicast application type.The IPv4 source address, group address, and the VPN route distinguisher are also specified. • *type-number* --the type-number. Valid values are from 0-65535.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables filtering of MLDP debugging output using the opaque type. This output occurs when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network. The opaque type represents the type of multicast application (for example, MVPN) occurring across an MPLS core network.

Examples

The following example shows how to enable filtering of MLDP debugging output using the opaque type:

```
Router# debug mpls mldp filter opaque_type mdt 100:2 0
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp generic

To enable debugging output for generic Multicast Label Distribution Protocol (MLDP) events, use the **debug mpls mldp generic**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp generic [**i** *d*]
no debug mpls mldp generic [**i** *d*]

Syntax Description

id	(Optional) The hexadecimal Label Switched Multicast (LSM) system ID.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for generic MLDP events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

Examples

The following example shows how to enable debugging output for generic MLDP events:

```
Router# debug mpls mldp generic
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp gr

To enable debugging output for Multicast Label Distribution Protocol (MLDP) graceful restart (GR) events, use the **debug mpls mldp gr**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp gr [*id*]
no debug mpls mldp [*id*]

Syntax Description

id	(Optional) The hexadecimal Label Switched Multicast (LSM) system ID.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for MLDP graceful restart events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

Examples

The following example shows how to enable debugging output for MLDP GR events:

```
Router# debug mpls mldp gr
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp mfi

To enable debugging output for Multicast Label Distribution Protocol/Multicast Forwarding Information (MLDP/MFI) interaction events, use the **debug mpls mldp mfi**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp mfi [*id*]
no debug mpls mldp mfi [*id*]

Syntax Description

id	(Optional) The hexadecimal Label Switched Multicast (LSM) system ID.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for MLDP/MFI interaction events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

Examples

The following example shows how to enable debugging output for MLDP/MFI interaction events:

```
Router# debug mpls mldp mfi
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp mrib

To enable debugging output for Multicast Label Distribution Protocol/Multicast Routing Information Base (MLDP/MRIB) interaction events, use the **debug mpls mldp mrib**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp mrib
no debug mpls mldp mrib

Syntax Description

This command has no arguments or keywords.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for MLDP/MRIB interaction events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

Examples

The following example shows how to enable debugging output for MLDP/MRIB interaction events:

```
Router# debug mpls mldp mrib
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp neighbor

To enable debugging output for Multicast Label Distribution Protocol (MLDP) neighbor events, use the **debug mpls mldp neighbor**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp neighbor
no debug mpls mldp neighbor

Syntax Description

This command has no arguments or keywords.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for MLDP neighbor events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN)core network.

Examples

The following example shows how to enable debugging output for MLDP neighbor events:

```
Router# debug mpls mldp neighbor
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp packet	Enables debugging output for MLDP-generated MPLS control plane events.
show mpls mldp database	Displays MLDP information.

debug mpls mldp packet

To enable debugging output for Multicast Label Distribution Protocol (MLDP)-generated Multiprotocol Label Switching (MPLS) control plane events, use the **debug mpls mldp packet**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls mldp packet
no debug mpls mldp packet

Syntax Description

This command has no arguments or keywords.

Command Default

The command is disabled by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command when the MLDP-based MVPN feature appears not to be functioning. The command enables debugging output for MLDP-generated MPLS control plane events that occur when the MLDP-based MVPN feature is enabled. This feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

Examples

The following example shows how to enable debugging output for MLDP-generated MPLS control plane events:

```
Router# debug mpls mldp packet
```

Related Commands

Command	Description
debug mpls mldp all	Enables debugging output for all MLDP events.
debug mpls mldp filter opaque_type	Enables filtering of MLDP debugging output using the opaque type.
debug mpls mldp generic	Enables debugging output for generic MLDP events.
debug mpls mldp gr	Enables debugging output for MLDP graceful restart events.
debug mpls mldp mfi	Enables debugging output for MLDP/MFI interaction events.
debug mpls mldp mrib	Enables debugging output for MLDP/MRIB interaction events.
debug mpls mldp neighbor	Enables debugging output for MLDP neighbor events.
show mpls mldp database	Displays MLDP information.

debug mpls netflow

To display debug messages for MPLS egress NetFlow accounting, use the **debug mpls netflow**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls netflow
no debug mpls netflow

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(10)ST	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXI4	This command was integrated into Cisco IOS Release 12.2(33)SXI4.

Examples

Here is sample output from the **debug mpls netflow** command:

```
Router# debug mpls netflow
MPLS Egress NetFlow debugging is on
Router#
Router#
Router#
4d00h:Egress flow:entry created, dest 3.3.3.3/32, src 34.0.0.1/8
Router#
Router#
4d00h:Egress flow:entry created, dest 3.3.3.3/32, src 42.42.42.42/32
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int eth1/4
Router(config-if)# no mpls netflow egress
Router(config-if)#
4d00h:MPLS output feature change, trigger TFIB scan
4d00h:tfib_scanner_walk, prefix 5.5.5.5/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 2.0.0.0/8, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 3.3.3.3/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 40.40.40.40/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 50.50.50.50/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 100.100.100.100/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 180.1.1.0/24, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 190.1.1.0/24, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 2.0.0.0/8, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 4.4.4.4/32, rewrite flow flag 1
```

```
4d00h:tfib_scanner_walk, prefix 40.40.40.40/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 50.50.50.50/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 177.1.1.0/24, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 180.1.1.0/24, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 190.1.1.0/24, rewrite flow flag 1
Router(config-if)#
Router(config-if)# mpls netflow egress
Router(config-if)#
4d00h:Interface refcount with output feature enabled = 2
4d00h:MPLS output feature change, trigger TFIB scan
4d00h:tfib_scanner_walk, prefix 5.5.5.5/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 2.0.0.0/8, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 3.3.3.3/32, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 40.40.40.40/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 50.50.50.50/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 100.100.100.100/32, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 180.1.1.0/24, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 190.1.1.0/24, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 2.0.0.0/8, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 4.4.4.4/32, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 40.40.40.40/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 50.50.50.50/32, rewrite flow flag 0
4d00h:tfib_scanner_walk, prefix 177.1.1.0/24, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 180.1.1.0/24, rewrite flow flag 1
4d00h:tfib_scanner_walk, prefix 190.1.1.0/24, rewrite flow flag 1
4d00h:Egress flow:entry created, dest 3.3.3.3/32, src 42.42.42.42/32
Router(config-if)#
Router(config-if)# end
Router# show run int eth1/4
Building configuration...
Current configuration:
!
interface Ethernet1/4
 ip vrf forwarding vpn1
 ip address 180.1.1.1 255.255.255.0
 no ip directed-broadcast
 mpls netflow egress
end
Router#
Router#
4d00h:%SYS-5-CONFIG_I:Configured from console by console
Router#
```

Note Flow flag 1 prefixes are reachable through this interface; therefore, MPLS egress NetFlow accounting is applied to all packets going out the destination prefix. Flow flag 0 prefixes are not reachable through this interface; therefore, MPLS egress NetFlow accounting is not applied to any packets going out the destination prefix.

Related Commands

Command	Description
show debug	Displays active debug output.

debug mpls packets

To display Multiprotocol Label Switching (MPLS) labeled packets switched by the host router, use the **debugmplspackets**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls packets [*interface*]
no debug mpls packets [*interface*]

Syntax Description

interface	(Optional) The interface or subinterface name.

Command Default

The debug output displays all labeled packets, regardless of the interface.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1CT	This command was introduced.
12.1(3)T	This command was modified for new MPLS terminology and syntax.
12.2(25)S	The command ouput was enhanced to display MPLS high availability information.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

The optional> interface parameter restricts the display to only those packets received or sent on the indicated interface or subinterface.

Note Use this command with care because it generates output for every packet processed. Furthermore, enabling this command causes fast and distributed label switching to be disabled for the selected interfaces. To avoid adversely affecting other system activity, use this command only when traffic on the network is at a minimum.

Examples

The following is sample output from the **debugmplspackets** command:

```
Router# debug mpls packets
TAG: Hs3/0: recvd: CoS=0, TTL=254, Tag(s)=27
TAG: Hs0/0: xmit: (no tag)
TAG: Hs0/0: recvd: CoS=0, TTL=254, Tag(s)=30
TAG: Hs3/0: xmit: CoS=0, TTL=253, Tag(s)=27
```

The following table describes the significant fields shown in the display.

Table 29: debug mpls packets Field Descriptions

Field	Description
Hs0/0	The identifier for the interface on which the packet was received or sent.
recvd	Packet received.
xmit	Packet transmitted.
CoS	Class of Service field from the packet label header.
TTL	Time to live field from the packet label header.
(no tag)	Last label was popped off the packet and sent unlabeled.
Tag(s)	A list of labels on the packet, ordered from the top of the stack to the bottom.

Cisco 10000 Series Example

The following is sample output from the **debugmplspackets** command:

```
Router# debug mpls packets
Gi6/0/0: rx: Len 118 Stack {30 6 255} - ipv4 data
Gi6/1/1: tx: Len 118 Stack {22 6 254} - ipv4 data
```

Related Commands

Command	Description
show mpls forwarding-table	Displays the contents of the MPLS forwarding table.

debug mpls static binding

To display information related to static binding events, use the **debug mpls static binding** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls static binding [**vrf** *vpn-name*]
no debug mpls static binding [**vrf** *vpn-name*]

Syntax Description

vrf *vpn-name*	(Optional) Displays information only for the specified VPN routing and forwarding instance.

Command Default

Static binding event information is not displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Usage Guidelines

Use the **debug mpls static binding** command to troubleshoot problems related to Multiprotocol Lbel Switching (MPLS) static labels or VRF-aware MPLS static labels. If you specify the **vrf** keyword, debugging is enabled only for a specified VRF. You can enable debugging only for a VRF.

Examples

The following is sample output from the **debug mpls static binding** command:

```
Router# debug mpls static binding vrf vpn100
MPLS Static label bindings debugging is on
00:15:13: mpls: Add remote static binding: 10.0.0.0/8; label 0; nexthop 172.16.0.8:0
00:15:13: mpls: Add static label binding for 10.0.0.0/8
00:15:13: mpls: Add static label binding for 10.0.0.1/8
00:15:13: mpls: Add remote static binding: 10.0.0.1/8; label 2607; nexthop 172.17.0.66:0
00:15:13: mpls: Add static label binding for 172.18.0.0/16
00:15:18: mpls: Periodic static label adjust
00:15:18: mpls: Static label update: 10.0.0.0/8
00:15:18:       Add remote label: nexthop: 172.16.0.8:0; label: 0
00:15:18: mpls: Periodic static label adjust
00:15:18: mpls: Static label update: 10.0.0.1/8
00:15:18:       Waiting for withdrawal of dynamic local label 55
00:15:18:       Add remote label: nexthop: 172.17.0.66:0; label: 2607
00:15:18: mpls: Periodic static label adjust
00:15:18: mpls: Static label update: 172.18.0.0/16
00:15:18:       Waiting for withdrawal of dynamic local label 17
```

```
00:15:28: mpls: Periodic static label adjust
00:15:28: mpls: Periodic static label adjust
00:15:28: mpls: Static label update: 10.0.0.1/8
00:15:28:       Local label 55 added to tib
00:15:28:       Signal route tag change, in label 55;out label 8388611; nh 192.168.44.77
00:15:28: mpls: Periodic static label adjust
00:15:28: mpls: Static label update: 172.18.0.0/16
00:15:28:       Local label 17 added to tib
00:15:28:       Signal route tag change, in label 17;out label 8388611; nh 192.168.44.66
00:15:38: mpls: Periodic static label adjust
00:15:38: mpls: Periodic static label adjust
00:15:38: mpls: Periodic static label adjust
```

Related Commands

Command	Description
mpls static binding ipv4 vrf	Binds a prefix to a local label.
show debug	Displays active debug output.

debug mpls tp

To enable debugging for Multiprotocol Label Switching (MPLS)-Transport Profile (TP), use the **debug mpls tp** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug mpls tp [{**all** | **cli** | **error** | **event** | **fault-oam** | **ha** | **init** | **link-num** | **lsp-db** | **lsp-ep** | **lsp-mp** | **mem** | **psc** | {**packet** | **event**} | **tun-db** | **tunnel**}]
no debug mpls tp

Syntax Description

all	(Optional) Displays all debug messages.
cli	(Optional) Displays MPLS-TP CLI debug messages.
error	(Optional) Displays MPLS-TP error debug messages.
event	(Optional) Displays MPLS-TP event debug messages.
fault-oam	(Optional) Displays MPLS-TP fault Operation, Administration, and Maintenance (OAM) messages.
ha	(Optional) Displays MPLS-TP high-availability debug messages.
init	(Optional) Displays MPLS-TP initialization debug messages.
link-num	(Optional) Displays MPLS-TP link-management debug messages
lsp-db	(Optional) Displays MPLS-TP midpoint link-state packet (LSP) database debug messages.
lsp-ep	(Optional) Displays MPLS-TP endpoint LSP debug messages.
lsp-mp	(Optional) Displays MPLS-TP midpoint LSP debug messages.
mem	(Optional) Displays MPLS-TP memory allocation and usage debug messages.
psc packet	(Optional) Displays MPLS packets received or transmitted by the Protection State Coordination (PSC) Protocol.
psc event	(Optional) Displays how the Protection State Coordination (PSC) Protocol behaves for any event it receives.
tun-db	(Optional) Displays MPLS-TP tunnel database debug messages.
tunnel	(Optional) Displays MPLS-TP tunnel debug messages.

Command Default

Debug messages are disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(1)SA	This command was introduced.

Release	Modification
15.1(3)S	This command was integrated into Cisco IOS Release 15.1(3)S.
Cisco IOS Release 3.9S	This command was modified. The **psc packet** and **event** keywords were added.

Examples

The following example enables the display of MPLS-TP endpoint LSP debug messages:

```
Router# debug mpls tp lsp-ep
debug mpls-tp endpoint lsp setup or use debugging is on
```

Related Commands

Command	Description
show mpls tp	Displays information summary or detailed information about MPLS-TP) settings.

debug mpls traffic-eng areas

To print information about traffic engineering area configuration change events, use the **debug mpls traffic-eng areas** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng areas
no debug mpls traffic-eng areas

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)ST	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about traffic engineering area configuration change events:

```
Router# debug mpls traffic-eng areas
TE-AREAS:isis level-1:up event
TE-PCALC_LSA:isis level-1
```

debug mpls traffic-eng autoroute

To print information about automatic routing over traffic engineering tunnels, use the **debug mpls traffic-eng autoroute** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng autoroute
no debug mpls traffic-eng autoroute

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)ST	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about automatic routing over traffic engineering tunnels:

```
Router# debug mpls traffic-eng autoroute
TE-Auto:announcement that destination 0001.0000.0003.00 has 1 tunnels
    Tunnel1 (traffic share 333, nexthop 10.112.0.12)
```

debug mpls traffic-eng auto-tunnel backup

To print system information about traffic engineering backup autotunnels, use the **debug mpls traffic-eng auto-tunnel backup** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng auto-tunnel backup [{**all** | **events** | **state**}]
no debug mpls traffic-eng auto-tunnel backup [{**all** | **events** | **state**}]

Syntax Description

all	(Optional) Enables all backup autotunnel debugging output.
events	(Optional) Prints backup autotunnel system events.
state	(Optional) Prints the system state of backup autotunnels.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(32)S	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.2(2)S	This command was modified. The output was enhanced to show debugging information for autotunnel and automesh stateful switchover (SSO) tunnels.
Cisco IOS XE Release 3.6S	This command was modified. The output was enhanced to show debugging information for autotunnel and automesh stateful switchover (SSO) tunnels.

Examples

This command shows how to display debugging information about the system state of backup autotunnels:

```
Router# debug mpls traffic-eng auto-tunnel backup state
```

Related Commands

Command	Description
debug mpls traffic-eng auto-tunnel primary	Prints system information about traffic engineering primary tunnels.
debug mpls traffic-eng tunnels events	Prints information about traffic engineering tunnel management system events.
mpls traffic-eng auto-tunnel backup	Automatically builds NHOP and NNHOP backup tunnels.

Command	Description
show ip explicit-paths	Displays the configured IP explicit paths.
show mpls traffic-eng tunnels	Displays information about tunnels.

debug mpls traffic-eng auto-tunnel primary

To print system information about traffic engineering primary autotunnels, use the **debug mpls traffic-eng auto-tunnel primary** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng auto-tunnel primary [{**all** | **events** | **state**}]
no debug mpls traffic-eng auto-tunnel primary [{**all** | **events** | **state**}]

Syntax Description

all	(Optional) Enables all primary autotunnel debugging output.
events	(Optional) Prints primary autotunnel system events.
state	(Optional) Prints the system state of primary autotunnels.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(32)S	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.2(2)S	This command was modified. The output was enhanced to show debugging information for autotunnel and automesh stateful switchover (SSO) tunnels.
Cisco IOS XE Release 3.6S	This command was modified. The output was enhanced to show debugging information for autotunnel and automesh stateful switchover (SSO) tunnels.

Examples

In the following example, debugging information is printed about system events of primary autotunnels:

```
Router# debug mpls traffic-eng auto-tunnel primary events

*Feb 6 18:12:57.871: TE_AUTO_TUN: primary CLI command:
interface tunnel3000
no logging event link-status
ip unnumbered Loopback0
tunnel destination 192.168.1.1
tunnel mode mpls traffic-eng
end
```

In the following example, debugging information is printed about the system state of primary autotunnels:

```
Router# debug mpls traffic-eng auto-tunnel primary state

Sample for debug mpls traffic-eng auto-tunnel primary state:  *Feb 6 18:11:44.363:
TE_AUTO_TUN: Didn't find protected Up Tunnel3000 to router id 192.168.1.1 out POS2/0
Sample for debug mpls traffic-eng auto-tunnel backup events (this is one log that prints
on multiple lines): *Feb 6 18:19:04.303: TE_AUTO_TUN: CLI command:
ip explicit-path name __dynamic_tunnel4000
index 1 next-address 192.168.1.2
```

Related Commands

Command	Description
debug mpls traffic-eng auto-tunnel backup	Prints system information about traffic engineering backup autotunnels.
debug mpls traffic-eng tunnels events	Prints information about traffic engineering tunnel management system events.
mpls mpls traffic-eng auto-tunnel primary config	Enables IP processing without an explicit address.
show ip explicit-paths	Displays the configured IP explicit paths.
show mpls traffic-eng tunnels	Displays information about tunnels.

debug mpls traffic-eng filter

To filter the display of Multiprotocol Label Switching (MPLS) traffic engineering messages by access control list (ACL), point-to-point (P2P) messages, or point-to-multipoint (P2MP) messages, use the **debug mpls traffic-eng filter** command in Privileged EXEC configuration mode. To disable the display of these messages, use the **no** form of this command.

debug mpls traffic-eng filter {**acl** *acl-num* | **dest-mode** {**p2p** | **p2mp**}}
no debug mpls traffic-eng filter {**acl** *acl-num* | **dest-mode** {**p2p** | **p2mp**}}

Syntax Description

acl *acl-num*	Displays debug information for the specified ACL. Valid values are 1-199 and 1300-2699.
dest-mode	Displays debug information sorted by P2P or P2MP messages.
p2p	Displays debug information about P2P tunnels.
p2mp	Displays debug information about P2MP tunnels.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRE	This command was introduced.

Examples

The following example enables the display of debug message for P2MP tunnels:

```
Router# debug mpls traffic-eng filter dest-mode p2mp
Setting filter for TE P2MP Tunnels/LSPs
```

Related Commands

Command	Description
show mpls traffic-eng tunnels	Displays information about P2P and P2MP tunnels.

debug mpls traffic-eng forwarding-adjacency

To display debug messages for traffic engineering (TE) forwarding adjacency events, use the **debug mpls traffic-eng forwarding-adjacency** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng forwarding-adjacency [**detail**] [*access-list-number*]
no debug mpls traffic-eng forwarding-adjacency

Syntax Description

detail	(Optional) Prints detailed debug information.
access-list-number	(Optional) Displays number of the access list. • A standard IP access list is in the range 1 to 199. • An extended IP access list is in the range 1300 to 2699.

Command Default

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(15)S	This command was introduced.
12.0(16)ST	This command was integrated into Cisco IOS Release 12.0(16)ST.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Use the **debug mpls traffic-eng forwarding-adjacency** command to troubleshoot any problems that occur after you configure the **tunnel mpls traffic-eng forwarding-adjacency** command.

If you enter the **detail** keyword before the *access-list-number* argument, you can specify an access list. However, if you enter an access list before you enter the **detail** keyword, you cannot specify the **detail** keyword.

Examples

The following is sample output from the **debug mpls traffic-eng forwarding-adjacency**command:

```
Router# debug mpls traffic-eng forwarding-adjacency
MPLS traffic-eng debugging is on
```

With a tunnel configured, the following output appears:

```
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 192.168.1.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng forwarding-adjacency
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng path-option 10 dynamic
end
```

When the tunnel comes up, the command generates the following debug output:

```
*Oct  2 12:27:07.846:TE-Auto:announcement that destination 0168.0001.0007.00 has 1 tunnels
*Oct  2 12:27:07.846:    Tunnel0     (traffic share 142857, nexthop 192.168.1.7)
*Oct  2 12:27:07.846:                (flags: Forward-Adjacency, holdtime 0)
```

Related Commands

Command	Description
show debug	Displays active debug output.
show mpls traffic-eng forwarding-adjacency	Displays TE tunnels being advertised as links in an IGP network.
tunnel mpls traffic-eng forwarding-adjacency	Advertises a TE tunnel as a link in an IGP network.

debug mpls traffic-eng ha sso

To display debugging output for Multiprotocol Label Switching (MPLS) traffic engineering high availability (HA) activities during the graceful switchover from an active Route Processor (RP) to a redundant standby RP, use the **debug mpls traffic-eng ha sso** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng ha sso {**auto-tunnel** | **errors** | **link-management** {**events** | **standby** | **recovery** | **checkpoint**} | **tunnel** {**events** | **standby** | **recovery**}}
no debug mpls traffic-eng ha sso {**auto-tunnel** | **errors** | **link-management** {**events** | **standby** | **recovery** | **checkpoint**} | **tunnel** {**events** | **standby** | **recovery**}}

Syntax Description

auto-tunnel	Displays information about autotunnel activity during the MPLS traffic engineering stateful switchover (SSO) process.
errors	Displays errors encountered during the MPLS traffic engineering SSO process.
link-management	Displays information about link management activity during the MPLS traffic engineering SSO process.
events	Displays significant events that occur during the MPLS traffic engineering SSO process.
standby	Displays information about the standby behavior during the MPLS traffic engineering SSO process.
recovery	Displays information about recovery activity during the MPLS traffic engineering SSO process.
checkpoint	Display information about checkpointing activities during the MPLS traffic engineering SSO process. Checkpointing occurs when a message is sent and acknowledged.
tunnel	Displays information about tunnel activity during the MPLS traffic engineering SSO process.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.2(2)S	This command was modified. The output for the **debug mpls traffic-eng ha sso tunnel events**, the **debug mpls traffic-eng ha sso standby**, and the **debug mpls traffic-eng ha sso tunnel recovery** commands was enhanced to show debugging information for autotunnel and automesh stateful switchover (SSO) tunnels.

Release	Modification
Cisco IOS XE Release 3.6S	This command was modified. The output for the **debug mpls traffic-eng ha sso tunnel events**, the **debug mpls traffic-eng ha sso standby**, and the **debug mpls traffic-eng ha sso tunnel recovery** commands was enhanced to show debugging information for autotunnel and automesh stateful switchover (SSO) tunnels.

Usage Guidelines

This command displays debugging output about the SSO process for MPLS traffic engineering tunnels, autotunnels, and link management systems. The SSO process occurs when the active router becomes unavailable and system control and routing protocol execution are transferred from the now inactive RP to the redundant standby RP, thus providing uninterrupted network services.

Examples

The following is sample output from the **debug mpls traffic-eng ha sso** command when you have enabled debugging keywords to monitor the SSO process for tunnels and link management systems as the standby router becomes active:

```
Router# debug mpls traffic-eng ha sso link-management events
MPLS traffic-eng SSO link management events debugging is on
Router# debug mpls traffic-eng ha sso link-management recovery
MPLS traffic-eng SSO link management recovery debugging is on
Router# debug mpls traffic-eng ha sso link-management standby
MPLS traffic-eng SSO link management standby behavior debugging is on
Router# debug mpls traffic-eng ha sso link-management
checkpoint
MPLS traffic-eng SSO link management checkpointed info debugging is on
Router# debug mpls traffic-eng ha sso tunnel standby
MPLS traffic-eng SSO tunnel standby behavior debugging is on
Router# debug mpls traffic-eng ha sso tunnel recovery
MPLS traffic-eng SSO tunnel head recovery debugging is on
Router# debug mpls traffic-eng ha sso tunnel events
MPLS traffic-eng SSO events for tunnel heads debugging is on
Router# debug mpls traffic-eng ha sso errors
MPLS traffic-eng SSO errors debugging is on
Router# show debug
<-----
This command displays the debugging that is enabled.
MPLS TE:
  MPLS traffic-eng SSO link management events debugging is on
  MPLS traffic-eng SSO link management recovery debugging is on
  MPLS traffic-eng SSO link management standby behavior debugging is on
  MPLS traffic-eng SSO link management checkpointed info debugging is on
  MPLS traffic-eng SSO tunnel standby behavior debugging is on
  MPLS traffic-eng SSO tunnel head recovery debugging is on
  MPLS traffic-eng SSO events for tunnel heads debugging is on
  MPLS traffic-eng SSO errors debugging is on
Router#
Standby-Router#
```

Following is the sample debugging output displayed during a successful SSO recovery on the standby router as it becomes active:

```
*May 12 20:03:15.303: RRR_HA_STATE: Told to wait for IGP convergence
*May 12 20:03:14.807: %FABRIC-SP-STDBY-5-FABRIC_MODULE_ACTIVE: The Switch Fabric Module in
 slot 5 became active.
*May 12 20:03:15.763: RRR_HA_REC: Attempting to recover last flooded info; protocol: OSPF,
 area: 0
*May 12 20:03:15.763: RRR_HA_REC: recovered ospf area 0 instance 0x48FFF240
```

```
*May 12 20:03:15.763: RRR_HA_REC: recovered system info
*May 12 20:03:15.763: RRR_HA_REC: recovered link[0] info
*May 12 20:03:15.763: RRR_HA: Recovered last flooded info for igp: OSPF, area: 0
*May 12 20:03:15.763: Pre announce tunnel 10
*May 12 20:03:15.763: TSPVIF_HA_EVENT: added Router_t10 to dest list
*May 12 20:03:15.763: TSPVIF_HA_EVENT: Completed announcement of 1 tunnel heads to IGP
*May 12 20:03:15.763: TSPVIF_HA_REC: Attempting to recover Tunnel10 after SSO
*May 12 20:03:15.763: LSP-TUNNEL-REOPT: Tunnel10 [61] set to recover
*May 12 20:03:15.763: TSPVIF_HA_REC: Recovered number hops = 5
*May 12 20:03:15.763: TSPVIF_HA_REC: recovered ospf area 0 instance 0x48FFF240
*May 12 20:03:15.763: TSPVIF_HA_REC: Recovered Hop 0: 10.0.3.1, Id: 10.0.0.3 Router Node
(ospf) flag:0x0
*May 12 20:03:15.763: TSPVIF_HA_REC: Recovered Hop 1: 10.0.3.2, Id: 10.0.0.7 Router Node
(ospf) flag:0x0
*May 12 20:03:15.763: TSPVIF_HA_REC: Recovered Hop 2: 10.0.6.1, Id: 10.0.0.7 Router Node
(ospf) flag:0x0
*May 12 20:03:15.763: TSPVIF_HA_REC: Recovered Hop 3: 10.0.6.2, Id: 10.0.0.9 Router Node
(ospf) flag:0x0
*May 12 20:03:15.763: TSPVIF_HA_REC: Recovered Hop 4: 10.0.0.9, Id: 10.0.0.9 Router Node
(ospf) flag:0x0
*May 12 20:03:15.763: TSPVIF_HA_REC: signalling recovered setup for Tunnel10: popt 1
[61], weight 2
*May 12 20:03:15.891: TSPVIF_HA_REC: recovered Tu10 forwarding info needed by query
*May 12 20:03:15.891: TSPVIF_HA_REC:     output_idb: GigabitEthernet3/2, output_nhop:
180.0.3.2
Standby-Router#
Router#
*May 12 20:03:25.891: TSPVIF_HA_REC: recovered Tu10 forwarding info needed by query
*May 12 20:03:25.891: TSPVIF_HA_REC:     output_idb: GigabitEthernet3/2, output_nhop:
10.0.3.2
*May 12 20:03:35.891: TSPVIF_HA_REC: recovered Tu10 forwarding info needed by query
*May 12 20:03:35.891: TSPVIF_HA_REC:     output_idb: GigabitEthernet3/2, output_nhop:
10.0.3.2
*May 12 20:03:35.895: RRR_HA_STATE: IGP flood prevented during IGP recovery
*May 12 20:03:38.079: LSP-TUNNEL-REOPT: Tunnel10 [61] received RESV for recovered setup
*May 12 20:03:38.079: LSP-TUNNEL-REOPT: Tunnel10 [61] removed as recovery
*May 12 20:03:38.079: TSPVIF_HA_EVENT: notifying RSVP HA to add lsp_info using key
10.0.0.3->10.0.0.9 Tu10 [61] 10.0.0.3
*May 12 20:03:38.079: TSPVIF_HA_EVENT: updated 7600-1_t10 state; action = add; result =
success
*May 12 20:03:38.079: TSPVIF_HA_EVENT: 7600-1_t10 fully recovered; rewrite refreshed
*May 12 20:03:38.079: TSPVIF_HA_EVENT: notifying CBTS bundle about Router_t10
*May 12 20:03:38.079: TSPVIF_HA_EVENT: notifying RSVP HA to remove lsp_info using key
10.0.0.3->10.0.0.9 Tu10 [61] 10.0.0.3
*May 12 20:03:38.079: RRR_HA: Received notification recovery has ended.  Notify IGP to
flood.
*May 12 20:03:38.079: TSPVIF_HA_EVENT: Received notification recovery has ended
*May 12 20:03:38.079: TSPVIF_HA_STANDBY: prevent verifying setups; IGP has not converged
*May 12 20:03:38.083: TSPVIF_HA_STANDBY: preventing new setups; reason: IGP recovering
*May 12 20:03:38.083: TSPVIF_HA_STANDBY: prevent verifying setups; IGP has not converged
*May 12 20:03:38.083: TSPVIF_HA_STANDBY: preventing new setups; reason: IGP recovering
*May 12 20:03:38.083: RRR_HA_STATE: IGP flood prevented during IGP recovery
7600-1#
*May 12 20:03:47.723: RRR_HA: Received notification that RIB table 0 has converged.
*May 12 20:03:47.723: RRR_HA: Received notification all RIBs have converged.  Notify IGP
to flood.
*May 12 20:03:47.723: RRR_HA_STATE: Told not to wait for IGP convergence
*May 12 20:03:47.723: RRR_HA_INFO: update flooded system info; action = add; result = success
*May 12 20:03:47.723: LM System key::
*May 12 20:03:47.723:   Flooding Protocol: ospf
*May 12 20:03:47.723:   IGP Area ID: 0
*May 12 20:03:47.723: LM Flood Data::
*May 12 20:03:47.723:   LSA Valid flags: 0x0  Node LSA flag: 0x0
*May 12 20:03:47.723:   IGP System ID: 10.0.0.3  MPLS TE Router ID: 10.0.0.3
```

```
*May 12 20:03:47.723:   Flooded links: 1  TLV length: 0 (bytes)
*May 12 20:03:47.723:   Fragment id: 0
*May 12 20:03:47.723: rrr_ha_lm_get_link_info_size: link size: 212 bytes; num TLVs: 0
*May 12 20:03:47.723: rrr_ha_sizeof_lm_link_info: link size: 212 bytes; num TLVs: 0
*May 12 20:03:47.723: RRR_HA_INFO: update flooded link[0] info; action = add;
result = success
*May 12 20:03:47.723: RRR HA Checkpoint Info Buffer::
*May 12 20:03:47.723:   Info Handle:          0x490BB1C8
*May 12 20:03:47.723:   Max Size:             212
*May 12 20:03:47.723:   Info Size:            212
*May 12 20:03:47.723:   Info Write Pointer:   0x490BB29C
*May 12 20:03:47.723: LM Link key::
*May 12 20:03:47.723:   Flooding Protocol: ospf  IGP Area ID: 0  Link ID: 0
(GigabitEthernet3/2)
*May 12 20:03:47.723:     Ifnumber: 5  Link Valid Flags: 0x193B
*May 12 20:03:47.723      Link Subnet Type: Broadcast
*May 12 20:03:47.723:     Local Intfc ID: 0  Neighbor Intf ID: 0
*May 12 20:03:47.723:     Link IP Address: 10.0.3.1
*May 12 20:03:47.723:     Neighbor IGP System ID: 10.0.3.2  Neighbor IP Address: 10.0.0.0
*May 12 20:03:47.723:     IGP Metric: 1  TE Metric: 1
*May 12 20:03:47.723:     Physical Bandwidth: 1000000 kbits/sec
*May 12 20:03:47.723:     Res. Global BW: 3000 kbits/sec
*May 12 20:03:47.723:     Res. Sub BW: 0 kbits/sec
*May 12 20:03:47.723:     Upstream::
Router#
*May 12 20:03:47.723:                                Global Pool   Sub Pool
*May 12 20:03:47.723:                                -----------   ----------
*May 12 20:03:47.723:     Reservable Bandwidth[0]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[1]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[2]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[3]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[4]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[5]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[6]:           0            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[7]:           0            0 kbits/sec
*May 12 20:03:47.723:     Downstream::
*May 12 20:03:47.723:                                Global Pool   Sub Pool
*May 12 20:03:47.723:                                -----------   ----------
*May 12 20:03:47.723:     Reservable Bandwidth[0]:        3000            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[1]:        3000            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[2]:        3000            0 kbits/sec
*May 12 20:03:47.723:     Reservable Bandwidth[3]:        3000            0 kbits/sec
*May 12 20:03:47.727:     Reservable Bandwidth[4]:        3000            0 kbits/sec
*May 12 20:03:47.727:     Reservable Bandwidth[5]:        3000            0 kbits/sec
*May 12 20:03:47.727:     Reservable Bandwidth[6]:        3000            0 kbits/sec
*May 12 20:03:47.727:     Reservable Bandwidth[7]:        2900            0 kbits/sec
*May 12 20:03:47.727:     Affinity Bits: 0x0
*May 12 20:03:47.727:     Protection Type: Capability 0,  Working Priority 0
*May 12 20:03:47.727:     Number of TLVs: 0
*May 12 20:03:47.727: RRR_HA: Updated flood state for ospf area 0 with 1 links); result =
success
Router#
```

The following example shows how to turn off debugging:

```
Router# no debug mpls traffic-eng ha sso link-management events
MPLS traffic-eng SSO link management events debugging is off
Router# no debug mpls traffic-eng ha sso link-management recovery
MPLS traffic-eng SSO link management recovery debugging is off
Router# no debug mpls traffic-eng ha sso link-management standby
MPLS traffic-eng SSO link management standby behavior debugging is off
Router# no debug mpls traffic-eng ha sso link-management checkpoint
MPLS traffic-eng SSO link management checkpointed info debugging is off
Router# no debug mpls traffic-eng ha sso tunnel standby
```

```
MPLS traffic-eng SSO tunnel standby behavior debugging is off
Router# no debug mpls traffic-eng ha sso tunnel recovery
MPLS traffic-eng SSO tunnel head recovery debugging is off
Router# no debug mpls traffic-eng ha sso tunnel events
MPLS traffic-eng SSO events for tunnel heads debugging is off
Router# no debug mpls traffic-eng ha errors
MPLS traffic-eng SSO errors debugging is off
```

Related Commands

Command	Description
debug ip rsvp high-availability	Displays debugging output for RSVP HA activities that improve the accessibility of network resources.
debug ip rsvp sso	Displays debugging output for RSVP activities during the graceful switchover from an active RP to a redundant RP.

debug mpls traffic-eng link-management admission-control

To print information about traffic engineering label-switched path (LSP) admission control on traffic engineering interfaces, use the **debug mpls traffic-eng link-management admission-control**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management admission-control [**detail**] [*acl-number*]
no debug mpls traffic-eng link-management admission-control [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information only for those LSPs that match the access list.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T, and the **detail** keyword and the *acl-number* argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about traffic engineering LSP admission control on traffic engineering interfaces:

```
Router# debug mpls traffic-eng link-management admission-control
TE-LM-ADMIT:tunnel 10.106.0.6 1_10002:created [total 4]
TE-LM-ADMIT:tunnel 10.106.0.6 1_10002: "None" -> "New"
TE-LM-ADMIT:tunnel 10.106.0.6 1_10002: "New" -> "Admitting 2nd Path Leg"
TE-LM-ADMIT:tunnel 10.106.0.6 1_10002: "Admitting 2nd Path Leg" -> "Path Admitted"
TE-LM-ADMIT:Admission control has granted Path query for 10.106.0.6 1_10002 (10.112.0.12)
on link Ethernet4/0/1 [reason 0]
TE-LM-ADMIT:tunnel 10.106.0.6 1_10002: "Path Admitted" -> "Admitting 1st Resv Leg"
TE-LM-ADMIT:tunnel 10.106.0.6 1_10002: "Admitting 1st Resv Leg" -> "Resv Admitted"
TE-LM-ADMIT:Admission control has granted Resv query for 10.106.0.6 1_10002 (10.112.0.12)
on link Ethernet4/0/1 [reason 0]
```

debug mpls traffic-eng link-management advertisements

To print information about resource advertisements for traffic engineering interfaces, use the **debugmplstraffic-englink-managementadvertisements** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management advertisements [**detail**] [*acl-number*]
no debug mpls traffic-eng link-management advertisements [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T. The **detail** keyword and the *acl-number*argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about resource advertisements for traffic engineering interfaces:

```
Router# debug mpls traffic-eng link-management advertisements detail
TE-LM-ADV:area isis level-1:IGP announcement:link Et4/0/1:info changed
TE-LM-ADV:area isis level-1:IGP msg:link Et4/0/1:includes subnet type (2), described nbrs
(1)
TE-LM-ADV:area isis level-1:IGP announcement:link Et4/0/1:info changed
TE-LM-ADV:area isis level-1:IGP msg:link Et4/0/1:includes subnet type (2), described nbrs
(1)
TE-LM-ADV:LSA:Flooding manager received message:link information change (Et4/0/1)
TE-LM-ADV:area isis level-1:*** Flooding node information ***
  System Information::
    Flooding Protocol:   ISIS
  Header Information::
    IGP System ID:       0001.0000.0001.00
    MPLS TE Router ID:   10.106.0.6
    Flooded Links:       1
  Link ID:: 0
    Link IP Address:     10.1.0.6
    IGP Neighbor:        ID 0001.0000.0001.02
    Admin. Weight:       10
```

```
    Physical Bandwidth:  10000 kbits/sec
    Max Reservable BW:   5000 kbits/sec
    Downstream::
      Reservable Bandwidth[0]:       5000 kbits/sec
      Reservable Bandwidth[1]:       2000 kbits/sec
      Reservable Bandwidth[2]:       2000 kbits/sec
      Reservable Bandwidth[3]:       2000 kbits/sec
      Reservable Bandwidth[4]:       2000 kbits/sec
      Reservable Bandwidth[5]:       2000 kbits/sec
      Reservable Bandwidth[6]:       2000 kbits/sec
Attribute Flags:  0x00000000
```

The following table describes the significant fields shown in the display.

Table 30: debug mpls traffic-eng link-management advertisements Field Descriptions

Field	Description
Flooding Protocol	Interior Gateway Protocol (IGB) that is flooding information for this area.
IGP System ID	Identification that IGP flooding uses in this area to identify this node.
MPLS TE Router ID	MPLS traffic engineering router ID.
Flooded Links	Number of links that are flooded in this area.
Link ID	Index of the link that is being described.
Link IP Address	Local IP address of this link.
IGP Neighbor	IGP neighbor on this link.
Admin. Weight	Administrative weight associated with this link.
Physical Bandwidth	Link’s bandwidth capacity (in kbps).
Max Reservable BW	Maximum amount of bandwidth that is currently available for reservation at this priority.
Reservable Bandwidth	Amount of bandwidth that is available for reservation.
Attribute Flags	Attribute flags of the link being flooded.

debug mpls traffic-eng link-management bandwidth-allocation

To print detailed information about bandwidth allocation for traffic engineering label-switched paths (LSPs), use the **debug mpls traffic-eng link-management bandwidth-allocation** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management bandwidth-allocation [**detail**] [*acl-number*]
no debug mpls traffic-eng link-management bandwidth-allocation [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information only for those LSPs that match the access list.

Command Default

No default behavior or values.

Command Modes

PrivilegedEXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T. The **detail** keyword and the *acl-number*argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about bandwidth allocation for traffic engineering LSPs:

```
Router# debug mpls traffic-eng link-management bandwidth-allocation
TE-LM-BW:tunnel 10.106.0.6 1_10002:requesting Downstream bw hold (3000000 bps [S]) on link
 Et4/0/1
TE-LM-BW:tunnel 10.106.0.6 1_10002:Downstream bw hold request succeeded
TE-LM-BW:tunnel 10.106.0.6 1_10002:requesting Downstream bw lock (3000000 bps [S]) on link
 Et4/0/1
TE-LM-BW:tunnel 10.106.0.6 1_10002:Downstream bw lock request succeeded∞_„Rs
```

Related Commands

Command	Description
debug mpls traffic-eng link-management admission-control	Prints information about traffic engineering LSP admission control on traffic engineering interfaces.
debug mpls traffic-eng link-management errors	Prints information about errors encountered during any traffic engineering link management procedure.

debug mpls traffic-eng link-management errors

To print information about errors encountered during any traffic engineering link management procedure, use the **debug mpls traffic-eng link-management errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management errors [**detail**]
no debug mpls traffic-eng link-management errors [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about errors encountered during a traffic engineering link management procedure:

```
Router# debug mpls traffic-eng link-management errors detail
00:04:48 TE-LM-ROUTING: link Et1/1/1: neighbor 0010.0000.0012.01: add to IP peer db failed
```

Related Commands

Command	Description
debug mpls traffic-eng link-management admission-control	Prints information about traffic engineering LSP admission control on traffic engineering interfaces.
debug mpls traffic-eng link-management advertisements	Prints information about resource advertisements for traffic engineering interfaces.
debug mpls traffic-eng link-management bandwidth-allocation	Prints information about bandwidth allocation for traffic engineering LSPs.
debug mpls traffic-eng link-management events	Prints information about traffic engineering link management system events.
debug mpls traffic-eng link-management igp-neighbors	Prints information about changes to the link management databases of IGP neighbors.

Command	Description
debug mpls traffic-eng link-management links	Prints information about traffic engineering link management interface events.

debug mpls traffic-eng link-management events

To print information about traffic engineering link management system events, use the **debug mpls traffic-eng link-management events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management events [**detail**]
no debug mpls traffic-eng link-management events [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T and the **detail** keyword was added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering link management system events:

```
Router# debug mpls traffic-eng link-management events detail
TE-LM-EVENTS:stopping MPLS TE Link Management process
TE-LM-EVENTS:MPLS TE Link Management process dying now
```

debug mpls traffic-eng link-management igp-neighbors

To print information about changes to the link management database of Interior Gateway Protocol (IGP) neighbors, use the **debug mpls traffic eng link-management igp-neighbors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management igp-neighbors [**detail**]
no debug mpls traffic-eng link-management igp-neighbors [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T and the **detail** keyword was added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about changes to the link management database of IGP neighbors:

```
Router# debug mpls traffic-eng link-management igp-neighbors detail
TE-LM-NBR:link AT0/0.2:neighbor 0001.0000.0002.00:created (isis level-1, 10.42.0.10, Up)[total
 2]
```

Related Commands

Command	Description
debug mpls traffic-eng link-management events	Prints information about traffic engineering-related ISIS events.

debug mpls traffic-eng link-management links

To print information about traffic engineering link management interface events, use the **debug mpls traffic-eng link-management links** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management links [**detail**]
no debug mpls traffic-eng link-management links [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T and the **detail** keyword was added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering link management interface events:

```
Router# debug mpls traffic-eng link-management links detail
TE-LM-LINKS:link AT0/0.2:RSVP enabled
TE-LM-LINKS:link AT0/0.2:increasing RSVP bandwidth from 0 to 5000000
TE-LM-LINKS:link AT0/0.2:created [total 2]
TE-LM-LINKS:Binding MPLS TE LM Admission Control as the RSVP Policy Server on ATM0/0.2
TE-LM-LINKS:Bind attempt succeeded
TE-LM-LINKS:link AT0/0.2:LSP tunnels enabled
```

debug mpls traffic-eng link-management preemption

To print information about traffic engineering label-switched path (LSP) preemption, use the **debug mpls traffic-eng link-management preemption** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management preemption [**detail**]
no debug mpls traffic-eng link-management preemption [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering LSP preemption:

```
Router# debug mpls traffic-eng link-management preemption detail
TE-LM-BW:preempting Downstream bandwidth, 1000000, for tunnel 10.106.0.6 2_2
TE-LM-BW:building preemption list to get bandwidth, 1000000, for tunnel 10.106.0.6 2_2
(priority 0)
TE-LM-BW:added bandwidth, 3000000, from tunnel 10.106.0.6 1_2 (pri 1) to preemption list
TE-LM-BW:preemption list build to get bw, 1000000, succeeded (3000000)
TE-LM-BW:preempting bandwidth, 1000000, using plist with 1 tunnels
TE-LM-BW:tunnel 10.106.0.6 1_2:being preempted on AT0/0.2 by 10.106.0.6 2_2
TE-LM-BW:preemption of Downstream bandwidth, 1000000, succeeded
```

debug mpls traffic-eng link-management routing

To print information about traffic engineering link management routing resolutions that can be performed to help Resource Reservation Protocol (RSVP) interpret explicit route objects, use the **debug mpls traffic-eng link-management routing** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng link-management routing [**detail**]
no debug mpls traffic-eng link-management routing [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T and the **detail** keyword was added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering link management routing resolutions that can be performed to help RSVP interpret explicit route objects:

```
Router# debug mpls traffic-eng link-management routing detail
TE-LM-ROUTING:route options to 10.42.0.10:building list (w/ nhop matching)
TE-LM-ROUTING:route options to 10.42.0.10:adding {AT0/0.2, 10.42.0.10}
TE-LM-ROUTING:route options to 10.42.0.10:completed list has 1 links
```

Related Commands

Command	Description
debug ip rsvp	Prints information about RSVP signalling events.

debug mpls traffic-eng load-balancing

To print information about unequal cost load balancing over traffic engineering tunnels, use the **debug mpls traffic-eng load-balancing** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng load-balancing
no debug mpls traffic-eng load-balancing

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)ST	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about unequal cost load balancing over traffic engineering tunnels:

```
Router# debug mpls traffic-eng load-balancing
TE-Load:10.210.0.0/16, 2 routes, loadbalancing based on MPLS TE bandwidth
TE-Load:10.200.0.0/16, 2 routes, loadbalancing based on MPLS TE bandwidth
```

debug mpls traffic-eng lsd-client

To display the Application Programming Interface (API) messages sent to the Label Switching Database (LSD) from the Traffic Engineering (TE) client, use the debug mpls traffic-eng lsd-client command in privileged EXEC mode. To disable the display of these messages, use the **no** form of this command.

debug mpls traffic-eng lsd-client
no debug mpls traffic-eng lsd-client

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(28)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(28)SXH.

Examples

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand enable TE globally:

```
00:10:23: TE-LSD-CLIENT: register with LSD OK; conn_id = 23, recov time = 60000 s
00:10:23: TE-LSD-CLIENT: LSD is now up
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand disable TE globally:

```
00:09:50: TE-LSD-CLIENT: unregister LSD client; result = OK; conn_id 23
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand enable TE on specific interfaces on Cisco 7500 series routers:

```
00:10:23: TE-LSD-CLIENT: enabled TE LSD client on Ethernet1/0; status = OK
00:10:23: TE-LSD-CLIENT: enabled TE LSD client on Serial2/0; status = OK
00:10:23: TE-LSD-CLIENT: enabled TE LSD client on Serial3/0; status = OK
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand disable TE on specific interfaces on Cisco 7500 series routers:

```
00:09:50: TE-LSD-CLIENT: disabled TE LSD client on Ethernet1/0; status = OK
00:09:50: TE-LSD-CLIENT: disabled TE LSD client on Serial2/0; status = OK
00:09:50: TE-LSD-CLIENT: disabled TE LSD client on Serial3/0; status = OK
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand enable TE on specific interfaces on Cisco 10000 series routers:

```
00:10:23: TE-LSD-CLIENT: enabled TE LSD client on GigabitEthernet1/0/0; status = OK
00:10:23: TE-LSD-CLIENT: enabled TE LSD client on Serial2/0/0; status = OK
00:10:23: TE-LSD-CLIENT: enabled TE LSD client on Serial3/0/0; status = OK
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand disable TE on specific interfaces on Cisco 10000 series routers:

```
00:09:50: TE-LSD-CLIENT: disabled TE LSD client on GigabitEthernet1/0/0; status = OK
00:09:50: TE-LSD-CLIENT: disabled TE LSD client on Serial2/0/0; status = OK
00:09:50: TE-LSD-CLIENT: disabled TE LSD client on Serial3/0/0; status = OK
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**command, allocate labels on tunnel midpoints, and create tunnel midpoint rewrites on Cisco 7500 series routers:

```
00:14:04: TE-LSD-CLIENT: label alloc OK; label = 16, conn_id = 23
00:14:04: TE-LSD-CLIENT: Create TE mid rewrite for 10.100.100.100 1 [5], Result: OK
00:14:04:          In: Serial3/0, 16 Out: Serial2/0, 3
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**command,allocate labels on tunnel midpoints, and create tunnel midpoint rewrites on a Cisco 10000 series router:

```
00:14:04: TE-LSD-CLIENT: label alloc OK; label = 16, conn_id = 23
00:14:04: TE-LSD-CLIENT: Create TE mid rewrite for 10.100.100.100 1 [5], Result: OK
00:14:04:          In: Serial3/0/0, 16 Out: Serial2/0/0, 3
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**command, free labels on tunnel midpoints, and delete tunnel midpoints on a Cisco 7500 series router:

```
00:13:13: TE-LSD-CLIENT: Delete TE mid rewrite for iou-100_t1, Result: OK
00:13:13: In: Serial3/0, 16 Out: Serial2/0, 1
00:13:13: TE-LSD-CLIENT: free label 16 result =  OK; conn_id = 23
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**command, free labels on tunnel midpoints, and delete tunnel midpoints on a Cisco 10000 series router:

```
00:13:13: TE-LSD-CLIENT: Delete TE mid rewrite for iou-100_t1, Result: OK
00:13:13: In: Serial3/0/0, 16 Out: Serial2/0/0, 1
00:13:13: TE-LSD-CLIENT: free label 16 result =  OK; conn_id = 23
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand create tunnel headend rewrites on a Cisco 7500 series router:

```
00:09:10: TE-LSD-CLIENT: Create TE he rewrite for iou-100_t1, Result = OK
00:09:10: tun_inst: 7  Out: Serial3/0, 16  Dest: 10.0.0.2
ps_flags: 0x60003
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand create tunnel headend rewrites on a Cisco 10000 series router:

```
00:09:10: TE-LSD-CLIENT: Create TE he rewrite for iou-100_t1, Result = OK
00:09:10: tun_inst: 7  Out: Serial3/0/0, 16  Dest: 10.0.0.2
ps_flags: 0x60003
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand delete tunnel headend rewrites on a Cisco 7500 series router:

```
00:09:15: TE-LSD-CLIENT: Delete TE he rewrite for iou-100_t1, Result: OK
00:09:15: tun_inst: 7  Out: Serial3/0, 16  ps_flags: 0x60003
```

The following messages are displayed when you issue the **debug mpls traffic-eng lsd-client**commandand delete tunnel headend rewrites on a Cisco 10000 series router:

```
00:09:15: TE-LSD-CLIENT: Delete TE he rewrite for iou-100_t1, Result: OK
00:09:15: tun_inst: 7  Out: Serial3/0/0, 16  ps_flags: 0x60003
```

Related Commands

Command	Description
debug mpls ip iprm events	Displays events related to the MPLS IPRM.
debug mpls ip iprm ldm	Displays debugging information for interactions between the IP LDMs and the MPLS IPRM.
debug mpls ip iprm mfi	Displays debugging information for interactions between the MFI and the MPLS IPRM.

debug mpls traffic-eng path

To display information about Multiprotocol Label Switching (MPLS) traffic engineering path calculation, use the **debug mpls traffic-eng path** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng path {**api** | **dump** | **errors***num* | **lookup** | **spf** | **verify**}
no debug mpls traffic-eng path {**api** | **dump** | **errors***num* | **lookup** | **spf** | **verify**}

Syntax Description

api	Displays path calculation application programming interface (API) events.
dump	Displays detailed path calculation information.
errors	Displays path calculation error event information.
num	The specific tunnel for which path calculation information is displayed. Valid values are 0-65535.
lookup	Displays information for path lookup events.
spf	Displays information for shortest path first (SPF) calculations.
verify	Displays information for path verifications.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)ST	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SRE	This command was modified. The **api**, **dump**, and **errors**keywords were added.

Examples

In the following example, information is printed about the calculation of the traffic engineering path:

```
Router# debug mpls traffic-eng path lookup
TE-PCALC:Tunnel1000 Path Setup to 10.110.0.10:FULL_PATH
TE-PCALC:bw 0, min_bw 0, metric:0
TE-PCALC:setup_pri 0, hold_pri 0
TE-PCALC:affinity_bits 0x0, affinity_mask 0xFFFF
TE-PCALC_PATH:create_path_hoplist:ip addr 10.42.0.6 unknown.
```

debug mpls traffic-eng process-restart

To display information about process restarts for reporting to your technical support representative, use the **debug mpls traffic-eng process-restart** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng process-restart
no debug mpls traffic-eng process-restart

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Usage Guidelines

If you report a problem and the **show mpls traffic-eng process-restart iprouting** displays abnormal results, your technical support representative might ask you to issue the **debug mpls traffic-eng process-restart** command, then perform an IP routing process restart and capture the output for analysis.

Examples

The following example shows partial output from an IP routing process restart:

```
Router# debug mpls traffic-eng process-restart
02:24:22: SM: ---TE ION Process Restart 0x78EF9050: process restart (3)
02:24:22: SM:    NORM (1) --> AWAIT-CFG (3)
02:24:22: TE ION Restart timer started, proc_idx:0 delay:120000
02:24:22: SM: ---TE ION Process Restart 0x78EF9050: process cfg replay start (4)
02:24:22: SM:    AWAIT-CFG (3) --> CFG (4)
02:24:22: TE ION Restart timer started, proc_idx:0 delay:300000
02:24:22: SM: ---TE ION Process Restart 0x78EF9050: reg invoke succeeded (2)
02:24:22: SM:    CFG (4) --> CFG (4)
02:24:22: SM: ---TE ION Process Restart 0x78EF9050: process cfg replay done (5)
02:24:22: SM:    CFG (4) --> SYNC (5)
02:24:22: TE ION Restart timer started, proc_idx:0 delay:900000
```

The output shows typical process restart information that your technical support representative might request if you report a problem after an IP routing process restart. The information displayed can vary, depending on the conditions that caused the restart.

Related Commands

Command	Description
show mpls traffic-eng process-restart iprouting	Displays the status of IP routing and MPLS traffic engineering synchronization after an IP routing process restarts.

debug mpls traffic-eng topology change

To print information about traffic engineering topology change events, use the **debug mpls traffic-eng topology change** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng topology change
no debug mpls traffic-eng topology change

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)ST	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about traffic engineering topology change events:

```
Router# debug mpls traffic-eng topology change
TE-PCALC_LSA:NODE_CHANGE_UPDATE isis level-1
        link flags:LINK_CHANGE_BW
        system_id:0001.0000.0001.00, my_ip_address:10.42.0.6
        nbr_system_id:0001.0000.0002.00, nbr_ip_address 10.42.0.10
```

debug mpls traffic-eng topology lsa

To print information about traffic engineering topology link state advertisement (LSA) events, use the **debug mpls traffic-eng topology lsa** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng topology lsa
no debug mpls traffic-eng topology lsa

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)ST	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, information is printed about traffic engineering topology LSA events:

```
Router# debug mpls traffic-eng topology lsa
TE-PCALC_LSA:node_lsa_add:Received a LSA:flags 0x1 !
IGP Id:0001.0000.0001.00, MPLS TE Id:10.106.0.6 is VALID has 2 links (frag_id 0)
      link[0 ]:Nbr IGP Id:0001.0000.0001.02
          frag_id 0, Intf Address:0.0.0.0
          admin_weight:10, attribute_flags:0x0
      link[1 ]:Nbr IGP Id:0001.0000.0002.00
          frag_id 0, Intf Address:10.42.0.6, Nbr Intf Address:10.42.0.10
          admin_weight:100, attribute_flags:0x0
TE-PCALC_LSA:(isis level-1):Received lsa:
IGP Id:0001.0000.0001.00, MPLS TE Id:10.106.0.6 Router Node  id 8
      link[0 ]:Nbr IGP Id:0001.0000.0002.00, nbr_node_id:9, gen:114
          frag_id 0, Intf Address:10.42.0.6, Nbr Intf Address:10.42.0.10
          admin_weight:100, attribute_flags:0x0
          physical_bw:155520 (kbps), max_reservable_bw:5000 (kbps)
               allocated_bw   reservable_bw      allocated_bw   reservable_bw
               ------------   -------------      ------------   -------------
        bw[0]:0              5000         bw[1]:3000           2000
        bw[2]:0              2000         bw[3]:0              2000
        bw[4]:0              2000         bw[5]:0              2000
        bw[6]:0              2000         bw[7]:0              2000
```

debug mpls traffic-eng tunnels errors

To print information about errors encountered during any traffic engineering tunnel management procedure, use the **debug mpls traffic-eng tunnels errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels errors [**detail**]
no debug mpls traffic-eng tunnels errors [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about errors encountered during a traffic engineering tunnel management procedure:

```
Router# debug mpls traffic-eng tunnels errors

00:04:14: LSP-TUNNEL-SIG: Tunnel10012[1]: path verification failed (unprotected) [Can't use
 link 10.12.4.4 on node 10.0.0.4]
```

debug mpls traffic-eng tunnels events

To print information about traffic engineering tunnel management system events, use the **debug mpls traffic-eng tunnels events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels events [**detail**]
no debug mpls traffic-eng tunnels events [**detail**]

Syntax Description

detail	(Optional) Prints detailed debugging information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T and the **detail** keyword was added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering tunnel management system events:

```
Router# debug mpls traffic-eng tunnels events detail
LSP-TUNNEL:received event:interface admin. down [Ethernet4/0/1]
LSP-TUNNEL:posting action(s) to all-tunnels:
              check static LSPs
LSP-TUNNEL:scheduling pending actions on all-tunnels
LSP-TUNNEL:applying actions to all-tunnels, as follows:
              check static LSPs
```

debug mpls traffic-eng tunnels labels

To print information about Multiprotocol Label Switching (MPLS) label management for traffic engineering tunnels, use the **debug mpls traffic-eng tunnels labels** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels labels [**detail**] [*acl-number*]
no debug mpls traffic-eng tunnels labels [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information only about traffic engineering tunnels that match the access list.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T, and the **detail** keyword and the *acl-number*argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about MPLS label management for traffic engineering tunnels:

```
Router# debug mpls traffic-eng tunnels labels detail
LSP-TUNNEL-LABELS:tunnel 10.106.0.6 1 [2]:fabric PROGRAM request
LSP-TUNNEL-LABELS:tunnel 10.106.0.6 1 [2]:programming label 16 on output interface ATM0/0.2
LSP-TUNNEL-LABELS:descriptor 71FA64:continuing "Program" request
LSP-TUNNEL-LABELS:descriptor 71FA64:set "Interface Point Out State" to, allocated
LSP-TUNNEL-LABELS:# of resource points held for "default" interfaces:2
LSP-TUNNEL-LABELS:descriptor 71FA64:set "Fabric State" to, enabled
LSP-TUNNEL-LABELS:descriptor 71FA64:set "Fabric Kind" to, default (LFIB)
LSP-TUNNEL-LABELS:descriptor 71FA64:set "Fabric State" to, set
LSP-TUNNEL-LABELS:tunnel 10.106.0.6 1 [2]:fabric PROGRAM reply
```

To restrict output to information about a single tunnel, you can configure an access list and supply it to the **debug** command. Configure the access list as follows:

```
Router(config-ext-nacl)# permit udp host scr_address host dst_address eq tun intfc
```

For example, if tunnel 10012 has destination 10.0.0.11 and source 10.0.0.4, as determined by the **show mpls traffic-eng tunnels command**, the following access list could be configured and added to the **debug** command:

```
Router(config-ext-nacl)# permit udp host 10.0.0.4 10.0.0.11 eq 10012
```

debug mpls traffic-eng tunnels reoptimize

To print information about traffic engineering tunnel reoptimizations, use the **debug mpls traffic-eng tunnels reoptimize** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels reoptimize [**detail**] [*acl-number*]
no debug mpls traffic-eng tunnels reoptimize [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information about only those traffic engineering tunnel reoptimizations that match the access list.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T, and the **detail** keyword and the *acl-number*argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering tunnel reoptimizations that match access list number 101:

```
Router# debug mpls traffic-eng tunnels reoptimize detail 101
LSP-TUNNEL-REOPT:Tunnel1 curr option 2 (0x6175CF8C), activate new option 2
LSP-TUNNEL-REOPT:Tunnel1 new path:option 2 [10002], weight 20
LSP-TUNNEL-REOPT:Tunnel1 old path:option 2 [2], weight 110
LSP-TUNNEL-REOPT:Tunnel1 [10002] set as reopt
LSP-TUNNEL-REOPT:Tunnel1 path option 2 [10002] installing as current
LSP-TUNNEL-REOPT:Tunnel1 [2] removed as current
LSP-TUNNEL-REOPT:Tunnel1 [2] set to delayed clean
LSP-TUNNEL-REOPT:Tunnel1 [10002] removed as reopt
LSP-TUNNEL-REOPT:Tunnel1 [10002] set to current
```

debug mpls traffic-eng tunnels signalling

To print information about traffic engineering tunnel signalling operations, use the **debug mpls traffic-eng tunnels signalling** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels signalling [**detail**] [*acl-number*]
no debug mpls traffic-eng tunnels signalling [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information about only those traffic engineering tunnel signalling operations that match the access list.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T, and the **detail** keyword and the *acl-number*argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering tunnel signalling operations that match access list number 101:

```
Router# debug mpls traffic-eng tunnels signalling detail 101
LSP-TUNNEL-SIG:tunnel Tunnel1 [2]:RSVP head-end open
LSP-TUNNEL-SIG:tunnel Tunnel1 [2]:received Path NHOP CHANGE
LSP-TUNNEL-SIG:Tunnel1 [2]:first hop change:0.0.0.0 --> 10.1.0.10
LSP-TUNNEL-SIG:received ADD RESV request for tunnel 10.106.0.6 1 [2]
LSP-TUNNEL-SIG:tunnel 10.106.0.6 1 [2]:path next hop is 10.1.0.10 (Et4/0/1)
LSP-TUNNEL-SIG:Tunnel1 [2] notified of new label information
LSP-TUNNEL-SIG:sending ADD RESV reply for tunnel 10.106.0.6 1 [2]
```

debug mpls traffic-eng tunnels state

To print information about state maintenance for traffic engineering tunnels, use the **debug mpls traffic-eng tunnels state** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels state [**detail**] [*acl-number*]
no debug mpls traffic-eng tunnels state [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information about state maintenance for traffic engineering tunnels that match the access list.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about state maintenance for traffic engineering tunnels that match access list number 99:

```
Router# debug mpls traffic-eng tunnels state detail 99
LSP-TUNNEL:tunnel 10.106.0.6 1 [2]: "Connected" -> "Disconnected"
LSP-TUNNEL:Tunnel1 received event:LSP has gone down
LSP-TUNNEL:tunnel 10.106.0.6 1 [2]: "Disconnected" -> "Dead"
LSP-TUNNEL-SIG:Tunnel1:changing state from up to down
LSP-TUNNEL:tunnel 10.106.0.6 1 [2]: "Dead" -> "Connected"
```

debug mpls traffic-eng tunnels timers

To print information about traffic engineering tunnel timer management, use the **debug mpls traffic-eng tunnels timers** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls traffic-eng tunnels timers [**detail**] [*acl-number*]
no debug mpls traffic-eng tunnels timers [**detail**] [*acl-number*]

Syntax Description

detail	(Optional) Prints detailed debugging information.
acl-number	(Optional) Uses the specified access list to filter the debugging information. Prints information about traffic engineering tunnel timer management that matches the access list.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T, and the **detail** keyword and the *acl-number*argument were added.
12.0(10)ST	This command was integrated into Cisco IOS Release 12.0(10)ST.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

In the following example, detailed debugging information is printed about traffic engineering tunnel timer management:

```
Router# debug mpls traffic-eng tunnels timers detail
LSP-TUNNEL-TIMER:timer fired for Action Scheduler
LSP-TUNNEL-TIMER:timer fired for Tunnel Head Checkup
```

debug mpls vpn ha

To enable the display of Virtual Private Network (VPN) high availability (HA) debugging information, use the **debug mpls vpn ha**command in privileged EXEC mode. To disable the display of VPN HA debugging information, use the **no** form of this command.

debug mpls vpn ha
no debug mpls vpn ha

Syntax Description

This command has no arguments or keywords.

Command Default

VPN HA debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.2(31)SB	This command was integrated into Cisco IOS Release 12.2(31)SB.

Examples

The following example shows sample output from the debug mpls vpn ha command:

```
Router# debug mpls vpn ha
VPN HA debugging is on.
```

debug mpls xtagatm cross-connect

Note Effective Cisco IOS Release 12.4(20)T, the **debugmplsxtagatmcross-connect** command is not available in Cisco IOS software.

To display requests and responses for establishing and removing cross-connects on the controlled ATM switch, use the **debugmplsxtagatmcross-connect**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls xtagatm cross-connect
no debug mpls xtagatm cross-connect

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(4)T	This command was updated to reflect the Multiprotocol Label Switching (MPLS) Internet Engineering Task Force (IETF) terminology.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was removed.

Usage Guidelines

This command monitors requests to establish or remove cross-connects from XmplsATM interfaces to the Virtual Switch Interface (VSI) master, as well as the VSI master responses to these requests.

Note Use this command with care, because it generates output for each cross-connect operation performed by the label switch controller (LSC). In a network configuration with many label virtual circuits (LVCs), the volume of output generated can interfere with system timing and the proper operation of other router functions. Use this command only in situations in which the LVC setup or teardown rate is low.

Examples

The following is sample output from the **debugmplsxtagatmcross-connect** command:

```
Router# debug mpls xtagatm cross-connect
XTagATM: cross-conn request; SETUP, userdata 0x17, userbits 0x1, prec 7
        0xC0100 (Ctl-If) 1/32 <-> 0xC0200 (XTagATM0) 0/32
XTagATM: cross-conn response; DOWN, userdata 0x60CDCB5C, userbits 0x2, result
OK
        0xC0200 1/37 --> 0xC0300 1/37
```

The following table describes the significant fields shown in the display.

Table 31: debug mpls xtagatm cross-connect Field Descriptions

Field	Description
XTagATM	The source of the debugging message as an XmplsATM interface.
cross-conn	An indicator that the debugging message pertains to a cross-connect setup or teardown operation.
request	A request from an XmplsATM interface to the VSI master to set up or tear down a cross-connect.
response	Response from the VSI master to an XmplsATM interface that a cross-connect was set up or removed.
SETUP	A request for the setup of a cross-connect.
TEARDOWN	A request for the teardown of a cross-connect.
UP	The cross-connect is established.
DOWN	The cross-connect is not established.
userdata, userbits	Values passed with the request that are returned in the corresponding fields in the matching response.
prec	The precedence for the cross-connect.
result	The status of the completed request.
0xC0100 (Ctl-If) 1/32	Information about the interface: • One endpoint of the cross-connect is on the interface whose logical interface number is 0xC0100. • The interface is the VSI control interface. • The virtual path identifier (VPI) value at this endpoint is 1. • The virtual channel identifier (VCI) value at this end of the cross-connect is 32.
<->	The type of cross-connect (unidirectional or bidirectional).
0xC0200 (XTagATM0) 0/32	Information about the interface: • The other endpoint of the cross-connect is on the interface whose logical interface number is 0xC0200. • The interface is associated with XmplsATM interface 0. • The VPI value at this endpoint is 0. • The VCI value at this end of the cross-connect is 32.

Field	Description
->	The response pertains to a unidirectional cross-connect.

Related Commands

Command	Description
show xtagatm cross-connect	Displays information about remotely connected ATM switches.

debug mpls xtagatm errors

Note Effective with Cisco IOS Release 12.4(20)T, the **debug mpls xtagatm errors** command is not available in Cisco IOS software.

To display information about error and abnormal conditions that occur on XmplsATM interfaces, use the **debug mpls xtagatm errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls xtagatm errors
no debug mpls xtagatm errors

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(4)T	This command was updated to reflect the Multiprotocol Label Switching (MPLS) Internet Engineering Task Force (IETF) terminology.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was removed.

Usage Guidelines

Use the **debug mpls xtagatm errors** command to display information about abnormal conditions and events that occur on XmplsATM interfaces.

Examples

The following is sample output from the **debug mpls xtagatm errors** command:

```
Router# debug mpls xtagatm errors
XTagATM VC: XTagATM0 1707 2/352 (ATM1/0 1769 3/915): Cross-connect setup
failed NO_RESOURCES
```

This message indicates a failed attempt to set up a cross-connect for a terminating a virtual circuit (VC) on XmplsATM0. The reason for the failure was a lack of resources on the controlled ATM switch.

debug mpls xtagatm events

Note Effective with Cisco IOS Release 12.4(20)T, the**debugmplsxtagatmevents** command is not available in Cisco IOS software.

To display information about major events that occur on XmplsATM interfaces, not including events for specific XmplsATM virtual circuits (VCs) and switch cross-connects, use the **debugmplsxtagatmevents**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls xtagatm events
no debug mpls xtagatm events

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Command	Modification
12.0(5)T	This command was introduced.
12.2(4)T	This command was updated to reflect the Multiprotocol Label Switching (MPLS) Internet Engineering Task Force (IETF) terminology.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	This command was removed.

Usage Guidelines

Use the **debugmplsxtagatmevents** command to monitor major events that occur on XmplsATM interfaces. This command monitors events that pertain only to XmplsATM interfaces as a whole and does not include any events that pertain to individual XmplsATM VCs or individual switch cross-connects. The specific events that are monitored when the **debugmplsxtagatmevents** command is in effect include the following:

- Receiving asynchronous notifications that the VSI master sent through the external ATM application programming interface (ExATM API) to an XmplsATM interface.
- Resizing of the table that is used to store switch cross-connect information. This table is resized automatically as the number of cross-connects increases.
- Marking of XmplsATM VCs as stale when an XmplsATM interface shuts down, thereby ensuring that the stale interfaces are refreshed before new XmplsATM VCs can be created on the interface.

Examples

The following is sample output from the **debugmplsxtagatmevents** command:

```
Router# debug mpls xtagatm events
XTagATM: desired cross-connect table size set to 256
XTagATM: ExATM API intf event Up, port 0xA0100 (None)
XTagATM: ExATM API intf event Down, port 0xA0100 (None)
XTagATM: marking all VCs stale on XTagATM0
```

The following table describes the significant fields shown in the display.

Table 32: debug mpls xtagatm events Field Descriptions

Field	Description
XTagATM	The source of the debugging message.
desired cross-connect table size set to 256	The table of cross-connect information has been set to hold 256 entries. A single cross-connect table is shared among all XmplsATM interfaces. The cross-connect table is automatically resized as the number of cross-connects increases.
ExATM API	The information in the debug output pertains to an asynchronous notification sent by the Virtual Switch Interface (VSI) master to the XmplsATM driver.
event Up/Down	The specific event that was sent by the VSI master to the XmplsATM driver.
port 0xA0100 (None)	The event pertains to the VSI interface whose logical interface number is 0xA0100, and that this logical interface is not bound to an XmplsATM interface.
marking all VCs stale on XTagATM0	All existing XmplsATM VCs on interface XmplsATM0 are marked as stale, and that XmplsATM0 remains down until all of these VCs are refreshed.

debug mpls xtagatm vc

Note Effective with Cisco IOS Release 12.4(20)T, the **debugmplsxtagatmvc** command is not available in Cisco IOS software.

To display information about events that affect individual XmplsATM terminating virtual circuits (VCs), use the **debugmplsxtagatmvc**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mpls xtagatm vc
no debug mpls xtagatm vc

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(4)T	This command was updated to reflect the Multiprotocol Label Switching (MPLS) Internet Engineering Task Force (IETF) terminology.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	This command was removed.

Usage Guidelines

Use the **debugmplsxtagatmvc** command to display detailed information about all events that affect individual XmplsATM terminating VCs.

Note Use this command with care, because it results in extensive output when many XmplsATM VCs are set up or torn down. This output can interfere with system timing and normal operation of other router functions. Use the **debugmplsxtagatmvc** command only when a few XmplsATM VCs are created or removed.

Examples

The following is sample output from the **debugmplsxtagatmvc** command:

```
Router# debug mpls xtagatm vc
XTagATM VC: XTagATM1 18 0/32 (ATM1/0 0 0/0):  Setup,  Down --> UpPend
XTagATM VC: XTagATM1 18 0/32 (ATM1/0 88 1/32):  Complete,  UpPend --> Up
```

```
XTagATM VC: XTagATM1 19 1/33 (ATM1/0 0 0/0):  Setup,  Down --> UpPend
XTagATM VC: XTagATM0 43 0/32 (ATM1/0 67 1/84):  Teardown,  Up --> DownPend
```

The following table describes the significant fields shown in the display.

Table 33: debug mpls xtagatm vc Field Descriptions

Field	Description
XTagATM VC	The source of the debugging message.
XTagATM <ifnum>	The particular XmplsATM interface number for the terminating VC.
vcd vpi/vci	The virtual circuit descriptor (VCD) and virtual path identifier/virtual channel identifier (VPI/VCI) values for the terminating VC.
(ctl-if vcd vpi/vci)	The control interface, the VCD, and the VPI and VCI values for the private VC corresponding to the XmplsATM VC on the control interface.
Setup, Complete, Teardown	The name of the event that occurred for the indicated VC.
oldstate -> newstate	The state of the terminating VC before and after the processing of the event.

debug mpoa client

Note Effective with Cisco IOS Release 15.1M, the **debug mpoa client**command is not available in Cisco IOS software.

To display Multiprotocol over ATM (MPOA) client debug information, use the **debug mpoa client**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

MPOA : debug mpoa client commanddebug mpoa client commanddebug mpoa client {**all** | **data** | **egress** | **general** | **ingress** | **keep-alives** | **platform-specific**} [**name** *mpc-name*]
no debug mpoa client {**all** | **data** | **egress** | **general** | **ingress** | **keep-alives** | **platform-specific**} [**name** *mpc-name*]

Syntax Description

all	Displays debugging information for all MPC activity.
data	Displays debugging information for data plane activity only. This option applies only to routers.
egress	Displays debugging information for egress functionality only.
general	Displays general debugging information only.
ingress	Displays debugging information for ingress functionality only.
keep-alives	Displays debugging information for keep-alive activity only.
platform-specific	Displays debugging information for specific platforms only. This option applies only to the Catalyst 5000 series ATM module.
name *mpc-name*	(Optional) Specifies the name of the MPC with the specified name.

Command Default

Debugging is turned on for all MPOA Clients (MPCs).

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
15.1M	This command was removed.

Examples

The following shows how to turn on debugging for the MPC ip_mpc:

```
ATM# debug mpoa client all name ip_mpc
```

Related Commands

Command	Description
debug mpoa server	Displays information about the MPOA server.

debug mpoa server

Note Effective with Cisco IOS Release 15.1M, the **debug mpoa server**command is not available in Cisco IOS software.

To display information about the Multiprotocol over ATM (MPOA) server, use the **debug mpoa server** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

dMPOA : debug mpoa server commanddebug mpoa server commandebug mpoa server [**name** *mps-name*]
no dMPOA : debug mpoa server commanddebug mpoa server commandebug mpoa server [**name** *mps-name*]

Syntax Description

name *mps-name*	(Optional) Specifies the name of an MPOA server.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
15.1M	This command was removed.

Usage Guidelines

The **debug mpoa server**command optionally limits the output only to the specified MPOA Server (MPS).

Examples

The following turns on debugging only for the MPS named ip_mps:

```
Router# debug mpoa server name ip_mps
```

debug mrcp

To display debugging messages for Media Resource Control Protocol (MRCP) operations, use the **debug mrcp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mrcp {**all** | **api** | **detail** | **error** | **pmh** | **session** | **socket** | **state**}
no debug mrcp {**all** | **api** | **detail** | **error** | **pmh** | **session** | **socket** | **state**}

Syntax Description

all	Displays all MRCP debugging messages.
api	Displays messages between the application and the MRCP stack.
detail	Displays detailed MRCP version 2 (MRCP v2) debugging messages.
error	Displays MRCP error messages.
pmh	Displays protocol message handler (PMH) messages.
session	Displays messages about active MRCP sessions.
socket	Displays MRCP v2 socket debugging messages
state	Displays Finite State Machine (FSM) messages.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced on the Cisco 3640, Cisco 3660, Cisco AS5300, Cisco AS5350, and Cisco AS5400.
12.4(15)T	This command was modified to support MRCP v2. The **detail** and **socket** keywords were added.

Examples

The following example shows output from the **debug mrcp api**command:

```
Router# debug mrcp api
```

The first four lines show Real Time Streaming Protocol (RTSP) socket commands for Text-To-Speech (TTS) operations:

```
*Apr 17 16:31:16.323:mrcp_add_param:param:Kill-On-Barge-In:
*Apr 17 16:31:16.323:mrcp_add_param:param:Speech-Language:
*Apr 17 16:31:16.323:mrcp_add_param:param:Logging-Tag:
*Apr 17 16:31:16.323:mrcp_add_param:param:Content-Base:
*Apr 17 16:31:16.323:mrcp_create_session:same host/port
*Apr 17 16:31:16.323:mrcp_associate_call 5 10
*Apr 17 16:31:16.323:mrcp_associate_call 5 10
*Apr 17 16:31:16.323:mrcp_synth_speak 5
```

```
*Apr 17 16:31:16.323:mrcp_add_param:param:Content-Base:
*Apr 17 16:31:16.323:mrcp_recognizer_define_grammar 5
```

The following lines show RTSP socket commands for Automatic Speech Recognition (ASR) operations:

```
*Apr 17 16:31:16.323:mrcp_add_param:param:Confidence-Threshold:
*Apr 17 16:31:16.323:mrcp_add_param:param:Sensitivity-Level:
*Apr 17 16:31:16.323:mrcp_add_param:param:Speed-Vs-Accuracy:
*Apr 17 16:31:16.323:mrcp_add_param:param:Dtmf-Interdigit-Timeout:
*Apr 17 16:31:16.323:mrcp_add_param:param:Dtmf-Term-Timeout:
*Apr 17 16:31:16.323:mrcp_add_param:param:Dtmf-Term-Char:
*Apr 17 16:31:16.323:mrcp_add_param:param:No-Input-Timeout:
*Apr 17 16:31:16.323:mrcp_add_param:param:Logging-Tag:
*Apr 17 16:31:16.327:mrcp_add_param:param:Content-Base:
*Apr 17 16:31:16.327:mrcp_add_param:param:Recognizer-Start-Timers:
*Apr 17 16:31:16.327:mrcp_recognizer_start 5
*Apr 17 16:31:26.715:mrcp_add_param:param:Kill-On-Barge-In:
*Apr 17 16:31:26.715:mrcp_add_param:param:Speech-Language:
*Apr 17 16:31:26.715:mrcp_add_param:param:Logging-Tag:
*Apr 17 16:31:26.715:mrcp_add_param:param:Content-Base:
*Apr 17 16:31:26.715:mrcp_synth_speak 5
*Apr 17 16:31:30.451:mrcp_destroy_session 5 type:SYNTHESIZER
*Apr 17 16:31:30.451:mrcp_destroy_session 5 type:RECOGNIZER
```

The following examples show output from the **debug mrcp error** command:

```
Router# debug mrcp error
```

This output shows an error when the response from the server is incorrect:

```
*May  9 20:29:09.936:Response from 10.1.2.58:554 failed
*May  9 20:29:09.936:MRCP/1.0 71 422 COMPLETE
```

This output shows an error when the RTSP connection to the server fails:

```
*May  9 20:29:09.936:Connecting to 10.1.2.58:554 failed
```

This output shows an error when the recognize request comes out of sequence:

```
*May  9 20:29:09.936:act_idle_recognize:ignoring old recognize request
```

The following example shows output from the **debug mrcp pmh** command:

```
Router# debug mrcp pmh
*Apr 17 16:32:51.777:param:Kill-On-Barge-In: true
*Apr 17 16:32:51.777:param:Speech-Language: en-US
*Apr 17 16:32:51.777:param:Logging-Tag: 14:14
*Apr 17 16:32:51.777:param:Content-Base: http://server-asr/
*Apr 17 16:32:51.777:param:Content-Base: http://server-asr/
*Apr 17 16:32:51.777:param:Confidence-Threshold: 50
*Apr 17 16:32:51.781:param:Sensitivity-Level: 50
*Apr 17 16:32:51.781:param:Speed-Vs-Accuracy: 50
*Apr 17 16:32:51.781:param:Dtmf-Interdigit-Timeout: 10000
*Apr 17 16:32:51.781:param:Dtmf-Term-Timeout: 10000
*Apr 17 16:32:51.781:param:Dtmf-Term-Char: #
*Apr 17 16:32:51.781:param:No-Input-Timeout: 10000
*Apr 17 16:32:51.781:param:Logging-Tag: 14:14
*Apr 17 16:32:51.781:param:Content-Base: http://server-asr/
*Apr 17 16:32:51.781:param:Recognizer-Start-Timers: false
*Apr 17 16:32:51.877:GRAMMAR-CONTENT-HEADER
```

```
*Apr 17 16:32:51.877:Content-Type:application/grammar+xml
Content-Id:field2@field.grammar
Content-Length:356
*Apr 17 16:32:51.885:GRAMMAR-CONTENT-HEADER
*Apr 17 16:32:51.885:Content-Type:text/uri-list
Content-Length:30
*Apr 17 16:32:51.885:Total-Length=365
*Apr 17 16:32:51.885:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*Apr 17 16:32:51.885:RECOGNIZE  20 MRCP/1.0
Confidence-Threshold:50
Sensitivity-Level:50
Speed-Vs-Accuracy:50
Dtmf-Interdigit-Timeout:10000
Dtmf-Term-Timeout:10000
Dtmf-Term-Char:#
No-Input-Timeout:10000
Logging-Tag:14:14
Content-Base:http://server-asr/
Recognizer-Start-Timers:false



*Apr 17 16:32:51.885:Content-Type:text/uri-list
Content-Length:30
*Apr 17 16:32:51.885:session:field2@field.grammar
*Apr 17 16:32:51.885:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*Apr 17 16:32:51.889:SPEECH-MARKUP-TYPE-HEADER
*Apr 17 16:32:51.889:Content-Type:application/synthesis+ssml
Content-Length:126
*Apr 17 16:32:51.889:Total-Length=313
*Apr 17 16:32:51.889:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*Apr 17 16:32:51.889:SPEAK  18 MRCP/1.0
Kill-On-Barge-In:true
Speech-Language:en-US
Logging-Tag:14:14
Content-Base:http://server-asr/



*Apr 17 16:32:51.889:Content-Type:application/synthesis+ssml
Content-Length:126
*Apr 17 16:32:51.889:<?xml version="1.0"?><speak> Who do you want speak to?? Joe, Carl,
Alex?. And I am extending the length of the text</speak>
*Apr 17 16:32:51.889:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*Apr 17 16:32:51.925:mrcp_pmh_parse_response:Length:28
 Apr 17 16:32:51.925:mrcp_pmh_get_request_line:Line:MRCP/1.0 19 200 COMPLETE
*Apr 17 16:32:51.925:Request-tag:19 resp-code:200 Status:COMPLETE
*Apr 17 16:32:51.925:No Of Properties:0
*Apr 17 16:32:51.925:mrcp_process_recog_response:
*Apr 17 16:32:51.933:mrcp_pmh_parse_response:Length:31
 Apr 17 16:32:51.933:mrcp_pmh_get_request_line:Line:MRCP/1.0 20 200 IN-PROGRESS
*Apr 17 16:32:51.933:Request-tag:20 resp-code:200 Status:IN-PROGRESS
*Apr 17 16:32:51.933:No Of Properties:0
*Apr 17 16:32:51.933:mrcp_process_recog_response:
*Apr 17 16:32:53.413:mrcp_pmh_parse_response:Length:31
 Apr 17 16:32:53.413:mrcp_pmh_get_request_line:Line:MRCP/1.0 18 200 IN-PROGRESS
*Apr 17 16:32:53.413:Request-tag:18 resp-code:200 Status:IN-PROGRESS
*Apr 17 16:32:53.413:No Of Properties:0
*Apr 17 16:32:53.413:mrcp_process_synth_response:
*Apr 17 16:33:01.685:mrcp_pmh_parse_response:Length:100
```

```
 Apr 17 16:33:01.689:mrcp_pmh_get_event_line:Line:SPEAK-COMPLETE 18 COMPLETE MRCP/1.0
*Apr 17 16:33:01.689:Request-tag:18 resp-code:200 Status:COMPLETE
*Apr 17 16:33:01.689:No Of Properties:2
*Apr 17 16:33:01.689:mrcp_process_synth_events:
*Apr 17 16:33:01.689: COMPLETION-CAUSE:1
*Apr 17 16:33:01.689:mrcp_send_synth_app_response:
*Apr 17 16:33:01.689:mrcp_pmh_parse_response:Length:61
 Apr 17 16:33:01.689:mrcp_pmh_get_event_line:Line:START-OF-SPEECH 20 IN-PROGRESS MRCP/1.0
*Apr 17 16:33:01.689:Request-tag:20 resp-code:200 Status:IN-PROGRESS
*Apr 17 16:33:01.689:No Of Properties:1
*Apr 17 16:33:01.689:mrcp_process_recog_events:
*Apr 17 16:33:02.653:mrcp_pmh_parse_response:Length:815
 Apr 17 16:33:02.653:mrcp_pmh_get_event_line:Line:RECOGNITION-COMPLETE 20 COMPLETE MRCP/1.0
*Apr 17 16:33:02.653:Request-tag:20 resp-code:200 Status:COMPLETE
*Apr 17 16:33:02.653:No Of Properties:2
*Apr 17 16:33:02.653:mrcp_process_recog_events:
*Apr 17 16:33:02.653: COMPLETION-CAUSE:0
*Apr 17 16:33:02.653:mrcp_send_recog_app_response:
*Apr 17 16:33:02.661:param:Kill-On-Barge-In: true
*Apr 17 16:33:02.661:param:Speech-Language: en-US
*Apr 17 16:33:02.661:param:Logging-Tag: 14:14
*Apr 17 16:33:02.661:param:Content-Base: http://server-asr/
*Apr 17 16:33:02.665:SPEECH-MARKUP-TYPE-HEADER
*Apr 17 16:33:02.665:Content-Type:application/synthesis+ssml
Content-Length:57
*Apr 17 16:33:02.665:Total-Length=243
*Apr 17 16:33:02.665:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*Apr 17 16:33:02.665:SPEAK  22 MRCP/1.0
Kill-On-Barge-In:true
Speech-Language:en-US
Logging-Tag:14:14
Content-Base:http://server-asr/
```

```
*Apr 17 16:33:02.665:Content-Type:application/synthesis+ssml
Content-Length:57
*Apr 17 16:33:02.665:<?xml version="1.0"?><speak> You have joe mails</speak>
*Apr 17 16:33:02.665:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*Apr 17 16:33:02.833:mrcp_pmh_parse_response:Length:31
 Apr 17 16:33:02.833:mrcp_pmh_get_request_line:Line:MRCP/1.0 22 200 IN-PROGRESS
*Apr 17 16:33:02.833:Request-tag:22 resp-code:200 Status:IN-PROGRESS
*Apr 17 16:33:02.833:No Of Properties:0
*Apr 17 16:33:02.833:mrcp_process_synth_response:
*Apr 17 16:33:06.382:mrcp_pmh_parse_response:Length:98
 Apr 17 16:33:06.382:mrcp_pmh_get_event_line:Line:SPEAK-COMPLETE 22 COMPLETE MRCP/1.0
*Apr 17 16:33:06.382:Request-tag:22 resp-code:200 Status:COMPLETE
*Apr 17 16:33:06.382:No Of Properties:2
*Apr 17 16:33:06.382:mrcp_process_synth_events:
*Apr 17 16:33:06.382: COMPLETION-CAUSE:0
*Apr 17 16:33:06.382:mrcp_send_synth_app_response:
```

The following example shows output from the **debug mrcp session**command:

```
Router# debug mrcp session
*Apr 17 16:34:07.851:mrcp_create_session:
*Apr 17 16:34:07.851:mrcp_create_session:New SCB creation
*Apr 17 16:34:07.851:mrcp_create_svr_session_url:
*Apr 17 16:34:07.851:mrcp_create_session:
*Apr 17 16:34:07.851:mrcp_create_session:Already an  SCB is created for this call
*Apr 17 16:34:07.851:mrcp_process_events:event:LIB_CONNECT SYNTHESIZERCONN-STATUS=0
```

```
*Apr 17 16:34:07.855:mrcp_process_events:event:SPEAK SYNTHESIZER
*Apr 17 16:34:07.855:mrcp_process_events:event:SPEAK defered
*Apr 17 16:34:07.855:mrcp_process_events:event:LIB_CONNECT RECOGNIZERCONN-STATUS=0
*Apr 17 16:34:07.855:mrcp_process_events:event:DEFINE_GRAMMAR RECOGNIZER
*Apr 17 16:34:07.855:mrcp_process_events:event:DEFINE_GRAMMAR defered
*Apr 17 16:34:07.855:mrcp_process_events:event:LIB_CONNECT RECOGNIZERCONN-STATUS=0
*Apr 17 16:34:07.855:mrcp_process_events:event:RECOGNIZE RECOGNIZER
*Apr 17 16:34:07.855:mrcp_process_events:event:RECOGNIZE defered
*Apr 17 16:34:07.855:mrcp_response_handler:status=RTSPLIB_STATUS_SERVER_CONNECTED
*Apr 17 16:34:07.855:mrcp_process_events:event:LIB_CONNECTED SYNTHESIZERCONN-STATUS=4
*Apr 17 16:34:07.947:mrcp_response_handler:status=RTSPLIB_STATUS_RTP_RECORD_SETUP
*Apr 17 16:34:07.947:mrcp_process_events:event:RECOG_RTP_SETUP RECOGNIZER
*Apr 17 16:34:07.947:mrcp_process_defered_events:event:DEFINE_GRAMMAR
*Apr 17 16:34:07.947:mrcp_process_defered_events:event:RECOGNIZECONN-STATUS=2
*Apr 17 16:34:07.971:mrcp_response_handler:status=RTSPLIB_STATUS_RECORD_ASSOCIATED
*Apr 17 16:34:07.971:mrcp_response_handler:status=RTSPLIB_STATUS_RTP_PLAY_SETUP
*Apr 17 16:34:07.975:mrcp_process_events:event:RECOGNIZER_ASSOCIATED RECOGNIZER
*Apr 17 16:34:07.975:mrcp_process_events:event:SYNTH_RTP_SETUP SYNTHESIZER
*Apr 17 16:34:07.975:mrcp_process_defered_events:event:SPEAKCONN-STATUS=1
*Apr 17 16:34:07.975:mrcp_response_handler:status=RTSPLIB_STATUS_PLAY_ASSOCIATED
*Apr 17 16:34:07.975:mrcp_process_events:event:SYNTHESIZER_ASSOCIATED SYNTHESIZER
*Apr 17 16:34:08.007:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:08.019:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:08.059:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:17.611:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:17.611:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:17.611:mrcp_process_events:event:SPEECH_COMPLETE SYNTHESIZER
*Apr 17 16:34:17.611:mrcp_process_events:event:START_OF_SPEECH RECOGNIZER
*Apr 17 16:34:18.575:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:18.575:mrcp_process_events:event:RECOGNITION_COMPLETE RECOGNIZER
*Apr 17 16:34:18.583:mrcp_process_events:event:SPEAK SYNTHESIZER
*Apr 17 16:34:18.587:mrcp_response_handler:status=RTSPLIB_STATUS_PLAY_ASSOCIATED
*Apr 17 16:34:18.587:mrcp_process_events:event:SYNTHESIZER_ASSOCIATED SYNTHESIZER
*Apr 17 16:34:18.763:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:22.279:mrcp_response_handler:status=RTSPLIB_STATUS_RESP_OK
*Apr 17 16:34:22.283:mrcp_process_events:event:SPEECH_COMPLETE SYNTHESIZER
*Apr 17 16:34:22.307:mrcp_process_events:event:LIB_DESTROY SYNTHESIZERCONN-STATUS=12
*Apr 17 16:34:22.311:mrcp_process_events:event:LIB_DESTROY RECOGNIZERCONN-STATUS=12
*Apr 17 16:34:22.311:mrcp_response_handler:status=RTSPLIB_STATUS_CLEANUP
*Apr 17 16:34:22.315:mrcp_free_fsm:
*Apr 17 16:34:22.315:mrcp_free_scb:
*Apr 17 16:34:22.315:mrcp_create_session_history:scb=0x62C712F4
*Apr 17 16:34:22.315:mrcp_insert_session_history_record:current=0x62999544, callID=0x12
*Apr 17 16:34:22.315:mrcp_insert_session_history_record:count = 3
*Apr 17 16:34:22.315:mrcp_insert_session_history_record:starting history record deletion_timer
 of 10 minutes
```

The following example shows output from the **debug mrcp state** command:

```
Router# debug mrcp state
*Apr 17 16:35:25.141:mrcp_add_synthesizer_fsm:adding synthesizer fsm
*Apr 17 16:35:25.141:mrcp_add_connection_fsm:adding connection fsm
*Apr 17 16:35:25.141:mrcp_add_rtpsetup_fsm:adding rtpsetup fsm
*Apr 17 16:35:25.145:hash_get:  key=7
*Apr 17 16:35:25.145:mrcp_add_recognizer_fsm:adding recognizer fsm
*Apr 17 16:35:25.145:mrcp_add_connection_fsm:adding connection fsm
*Apr 17 16:35:25.145:mrcp_add_rtpsetup_fsm:adding rtpsetup fsm
*Apr 17 16:35:25.145:mrcp_fsm_execute:type=SYNTHESIZER
```

The following lines show the gateway connecting to the TTS server:

```
*Apr 17 16:35:25.145:    curr[CONNECT_IDLE]  ev-id[LIB_CONNECT]
    next[CONNECTING] action=610B8FD00
```

```
*Apr 17 16:35:25.145:act_idle_libconnect
*Apr 17 16:35:25.145:mrcp_shortcut_connection_fsm
*Apr 17 16:35:25.149:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:25.149:    curr[CONNECTING]  ev-id[LIB_CONNECT_PENDING]
    next[CONNECTING] action=610B90F80
*Apr 17 16:35:25.149:act_connecting_libpending
*Apr 17 16:35:25.149:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:25.149:    curr[CONNECTING]  ev-id[LIB_CONNECT]
    next[CONNECTING] action=610B8D480
*Apr 17 16:35:25.149:act_connectfsm_error
*Apr 17 16:35:25.149:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:25.149:    curr[CONNECTING]  ev-id[LIB_CONNECT]
```

The following lines show the gateway successfully connected to the TTS server:

```
    next[CONNECTING] action=610B8D480
*Apr 17 16:35:25.149:act_connectfsm_error
*Apr 17 16:35:25.149:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:25.149:    curr[CONNECTING]  ev-id[LIB_CONNECTED]
    next[CONNECTED] action=610B913C0
*Apr 17 16:35:25.149:act_connecting_libconnected
*Apr 17 16:35:25.149:act_rtpsetupfsm_libdescribed
*Apr 17 16:35:25.237:mrcp_fsm_execute:type=RESOURCE_NONE
*Apr 17 16:35:25.237:    curr[RTP_IDLE]  ev-id[RECOG_RTP_SETUP]
    next[RTP_RECOG_SETUP_DONE] action=610B94F40
*Apr 17 16:35:25.237:act_idle_recog_rtpsetup
*Apr 17 16:35:25.237:mrcp_fsm_execute:type=RECOGNIZER
*Apr 17 16:35:25.237:    curr[RECOG_IDLE]  ev-id[DEFINE_GRAMMAR]
    next[RECOG_IDLE] action=610B99340
*Apr 17 16:35:25.237:act_idle_define_grammar:
*Apr 17 16:35:25.237:hash_add:  key=31
*Apr 17 16:35:25.237:mrcp_fsm_execute:type=RECOGNIZER
*Apr 17 16:35:25.237:    curr[RECOG_IDLE]  ev-id[RECOGNIZE]
    next[RECOG_ASSOCIATING] action=610B98400
*Apr 17 16:35:25.237:act_idle_recognize:
*Apr 17 16:35:25.245:mrcp_fsm_execute:type=RECOGNIZER
*Apr 17 16:35:25.245:    curr[RECOG_ASSOCIATING]  ev-id[RECOGNIZER_ASSOCIATED]
    next[RECOGNIZING] action=610B9AB40
*Apr 17 16:35:25.245:act_associating_recognizer_associated:
*Apr 17 16:35:25.249:hash_add:  key=32
*Apr 17 16:35:25.249:mrcp_fsm_execute:type=RESOURCE_NONE
*Apr 17 16:35:25.249:    curr[RTP_IDLE]  ev-id[SYNTH_RTP_SETUP]
    next[RTP_SYNTH_SETUP_DONE] action=610B93D40
*Apr 17 16:35:25.249:act_idle_synth_rtpsetup
*Apr 17 16:35:25.249:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:25.249:    curr[SYNTH_IDLE]  ev-id[SPEAK]
    next[SYNTH_ASSOCIATING] action=610BA5540
*Apr 17 16:35:25.249:act_idle_speak
*Apr 17 16:35:25.249:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:25.249:    curr[SYNTH_ASSOCIATING]  ev-id[SYNTHESIZER_ASSOCIATED]
```

The following lines show the TTS server performing speech synthesis:

```
    next[SPEAKING] action=610BA7B40
*Apr 17 16:35:25.249:act_associating_speak_associated
*Apr 17 16:35:25.249:hash_add:  key=30
*Apr 17 16:35:25.285:hash_get:  key=31
*Apr 17 16:35:25.285:hash_delete:  key=31
*Apr 17 16:35:25.293:hash_get:  key=32
*Apr 17 16:35:25.293:hash_get:  key=30
*Apr 17 16:35:32.805:hash_get:  key=30
*Apr 17 16:35:32.805:hash_delete:  key=30
*Apr 17 16:35:32.805:mrcp_fsm_execute:type=SYNTHESIZER
```

```
*Apr 17 16:35:32.805:    curr[SPEAKING]  ev-id[SPEECH_COMPLETE]
    next[SYNTH_IDLE] action=610BAA680
*Apr 17 16:35:32.805:act_speaking_speech_complete
*Apr 17 16:35:32.809:hash_get:  key=32
*Apr 17 16:35:32.809:mrcp_fsm_execute:type=RECOGNIZER
*Apr 17 16:35:32.809:    curr[RECOGNIZING]  ev-id[START_OF_SPEECH]
    next[RECOGNIZING] action=610B9F3C0
*Apr 17 16:35:32.809:act_recognizing_start_of_speech
*Apr 17 16:35:33.781:hash_get:  key=32
*Apr 17 16:35:33.781:hash_delete:  key=32
*Apr 17 16:35:33.781:mrcp_fsm_execute:type=RECOGNIZER
*Apr 17 16:35:33.781:    curr[RECOGNIZING]  ev-id[RECOGNITION_COMPLETE]
    next[RECOGNIZED] action=610B9D240
*Apr 17 16:35:33.781:act_recognizing_recognition_complete:
*Apr 17 16:35:33.789:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:33.789:    curr[SYNTH_IDLE]  ev-id[SPEAK]
    next[SYNTH_ASSOCIATING] action=610BA5540
*Apr 17 16:35:33.789:act_idle_speak
*Apr 17 16:35:33.793:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:33.793:    curr[SYNTH_ASSOCIATING]  ev-id[SYNTHESIZER_ASSOCIATED]
    next[SPEAKING] action=610BA7B40
*Apr 17 16:35:33.793:act_associating_speak_associated
*Apr 17 16:35:33.793:hash_add:  key=34
*Apr 17 16:35:33.949:hash_get:  key=34
*Apr 17 16:35:37.221:hash_get:  key=34
*Apr 17 16:35:37.221:hash_delete:  key=34
*Apr 17 16:35:37.221:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:37.221:    curr[SPEAKING]  ev-id[SPEECH_COMPLETE]
    next[SYNTH_IDLE] action=610BAA680
*Apr 17 16:35:37.221:act_speaking_speech_complete
*Apr 17 16:35:37.245:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:37.249:    curr[CONNECTED]  ev-id[LIB_DESTROY]
    next[CONNECTED] action=610B8DD00
*Apr 17 16:35:37.249:act_connected_libdestroy
*Apr 17 16:35:37.249:mrcp_fsm_execute:type=SYNTHESIZER
*Apr 17 16:35:37.249:    curr[CONNECTED]  ev-id[LIB_DESTROY]
    next[CONNECTED] action=610B8DD00
*Apr 17 16:35:37.249:act_connected_libdestroy
```

The following example shows output from the **debug mrcp detail**command:

```
Router# debug mrcp detail
*Sep  1 21:37:53.652: //68//MRCP:/mrcpv2_allocate_scb:
   scb=0xC07318C8, root_scb=0x661BDD54
*Sep  1 21:37:53.708: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=FALSE;TotalLength=165
*Sep  1 21:37:53.708: //-1//MRCP:/MRCPV2_ADD_HEADER:
   TotalLength=87
*Sep  1 21:37:53.708: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=TRUE;TotalLength=535:
MRCP/2.0 535      DEFINE-GRAMMAR  1
Channel-Identifier: 0000251844F8ACAD@speechrecog
:
Speech-Language: en-US
Content-Base: http://http-server1/php/
:
Content-Type: application/srgs+xml
Content-Id: field24@field.grammar
Content-Length: 290
:
<?xml version="1.0"?><grammar mode="voice" version="1.0" root="xxx"
xmlns="http://www.w3.org/2001/06/grammar" xml:lang="en-US">
       <rule id="xxx" scope="public">
         <one-of>
```

```
           <item>one</item>
           <item>two</item>
         </one-of>
       </rule>
      </grammar>
*Sep  1 21:37:53.708: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=FALSE;TotalLength=160
*Sep  1 21:37:53.708: //-1//MRCP:/MRCPV2_ADD_HEADER:
   TotalLength=82
*Sep  1 21:37:53.708: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=TRUE;TotalLength=499:
MRCP/2.0 499      RECOGNIZE  2
Channel-Identifier: 0000251844F8ACAD@speechrecog
:
Speech-Language: en-US
Confidence-Threshold: 0.50
Sensitivity-Level: 0.50
Speed-Vs-Accuracy: 0.50
Dtmf-Interdigit-Timeout: 10000
Dtmf-Term-Timeout: 0
Dtmf-Term-Char: #
No-Input-Timeout: 20000
N-Best-List-Length: 1
Logging-Tag: 68:68
Content-Base: http://http-server1/php/
Media-Type: audio/basic
Start-Input-Timers: false
:
Content-Type: text/uri-list
Content-Length: 31
:
session:field24@field.grammar
:
MRCP/2.0 80 1 200 COMPLETE
Channel-Identifier: 0000251844F8ACAD@speechrecog
:
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 0000251844F8ACAD@speechrecog
*Sep  1 21:37:57.404: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=FALSE;TotalLength=169
*Sep  1 21:37:57.404: //-1//MRCP:/MRCPV2_ADD_HEADER_CR:
   TotalLength=93
*Sep  1 21:37:57.404: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=TRUE;TotalLength=93:
MRCP/2.0 93       START-INPUT-TIMERS  3
Channel-Identifier: 0000251844F8ACAD@speechrecog
:
MRCP/2.0 80 3 200 COMPLETE
Channel-Identifier: 0000251844F8ACAD@speechrecog
:
MRCP/2.0 148 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 0000251844F8ACAD@speechrecog
Proxy-Sync-Id: 0F1F813000000148
Input-Type: speech
:
MRCP/2.0 589 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 0000251844F8ACAD@speechrecog
Proxy-Sync-Id: 0F1F813000000148
Completion-Cause: 000 success
Content-Type: application/nlsml+xml
Content-Length: 369
<?xml version="1.0" encoding="UTF-8"?>
<result grammar="session:field24@field.grammar">
 <interpretation grammar="session:field24@field.grammar" confidence="0.646043">
```

```
  <instance confidence="0.646043">
   one
  </instance>
  <input mode="speech" confidence="0.646043">
   one
   <input confidence="0.646043">
    one
   </input>
  </input>
 </interpretation>
</result>
*Sep  1 21:37:59.588: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=FALSE;TotalLength=165
*Sep  1 21:37:59.588: //-1//MRCP:/MRCPV2_ADD_HEADER:
   TotalLength=87
*Sep  1 21:37:59.588: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=TRUE;TotalLength=566:
MRCP/2.0 566      DEFINE-GRAMMAR  1
Channel-Identifier: 00001FEC44F8AA93@speechrecog
:
Speech-Language: en-US
Content-Base: http://http-server1/php/
:
Content-Type: application/srgs+xml
Content-Id: field25@field.grammar
Content-Length: 321
:
<?xml version="1.0"?><grammar mode="voice" version="1.0" root="xxx"
xmlns="http://www.w3.org/2001/06/grammar" xml:lang="en-US">
       <rule id="xxx" scope="public">
         <one-of>
           <item>three</item>
           <item>four</item>
           <item>one</item>
         </one-of>
       </rule>
      </grammar>
*Sep  1 21:37:59.588: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=FALSE;TotalLength=160
*Sep  1 21:37:59.588: //-1//MRCP:/MRCPV2_ADD_HEADER:
   TotalLength=82
*Sep  1 21:37:59.588: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=TRUE;TotalLength=499:
MRCP/2.0 499      RECOGNIZE  2
Channel-Identifier: 00001FEC44F8AA93@speechrecog
:
Speech-Language: en-US
Confidence-Threshold: 0.50
Sensitivity-Level: 0.50
Speed-Vs-Accuracy: 0.50
Dtmf-Interdigit-Timeout: 10000
Dtmf-Term-Timeout: 0
Dtmf-Term-Char: #
No-Input-Timeout: 10000
N-Best-List-Length: 1
Logging-Tag: 68:68
Content-Base: http://http-server1/php/
Media-Type: audio/basic
Start-Input-Timers: false
:
Content-Type: text/uri-list
Content-Length: 31
:
session:field25@field.grammar
```

```
:
MRCP/2.0 80 1 200 COMPLETE
Channel-Identifier: 00001FEC44F8AA93@speechrecog
:
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 00001FEC44F8AA93@speechrecog
*Sep  1 21:38:00.044: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=FALSE;TotalLength=169
*Sep  1 21:38:00.044: //-1//MRCP:/MRCPV2_ADD_HEADER_CR:
   TotalLength=93
*Sep  1 21:38:00.044: //-1//MRCP:/MRCPV2_ADD_REQUEST_LINE:
   IsFinal=TRUE;TotalLength=93:
MRCP/2.0 93       START-INPUT-TIMERS  3
Channel-Identifier: 00001FEC44F8AA93@speechrecog
:
MRCP/2.0 80 3 200 COMPLETE
Channel-Identifier: 00001FEC44F8AA93@speechrecog
:
MRCP/2.0 148 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 00001FEC44F8AA93@speechrecog
Proxy-Sync-Id: 0925248800000011
Input-Type: speech
:
MRCP/2.0 589 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 00001FEC44F8AA93@speechrecog
Proxy-Sync-Id: 0925248800000011
Completion-Cause: 000 success
Content-Type: application/nlsml+xml
Content-Length: 369
<?xml version="1.0" encoding="UTF-8"?>
<result grammar="session:field25@field.grammar">
 <interpretation grammar="session:field25@field.grammar" confidence="0.701971">
  <instance confidence="0.701971">
   one
  </instance>
  <input mode="speech" confidence="0.701971">
   one
   <input confidence="0.701971">
    one
   </input>
  </input>
 </interpretation>
</result>
```

The following example shows output from the **debug mrcp socket**command:

```
Router# debug mrcp socket
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_tcp_socket_connect:
   Socket=0,Dest=10.1.2.201:51001
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_connect_to_server:
   SocketConnectStatus[MRCPV2_SOCK_CONNECT_PENDING(2)], SocketId=0,
   ServerSession=0xC0732278, Dest=10.1.2.201:51001
*Sep  1 21:52:58.392: //-1//MRCP:/mrcpv2_handle_socket_read:
   Before Execute: Socket=0, SocketStatus=MRCPV2_SOCK_CONNECT_PENDING(2)
*Sep  1 21:52:58.392: //-1//MRCP:/mrcpv2_handle_socket_read:
   After Execute: Socket=0, SocketStatus=MRCPV2_SOCK_CONNECTED(1)
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_partial_socket_send:
   (Socket:0 Length:87) 600 bytes of data
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_partial_socket_send:
   Buffer Sent Successfully; fd=0, Sent=87
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_partial_socket_send:
   (Socket:0 Length:64) 600 bytes of data
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_partial_socket_send:
   Buffer Sent Successfully; fd=0, Sent=64
```

```
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_partial_socket_send:
   (Socket:0 Length:94) 600 bytes of data
*Sep  1 21:52:58.392: //74//MRCP:/mrcpv2_partial_socket_send:
   Buffer Sent Successfully; fd=0, Sent=94
```

Related Commands

Command	Description
show mrcp client session active	Displays information about active MRCP sessions.
show mrcp client session history	Displays information about past MRCP sessions.
show mrcp client statistics hostname	Displays statistics about MRCP sessions.

debug mspi receive

Note Effective with release 12.3(8)T, the **debug mspi receive**command is replaced by the **debug fax mspi**command. See the **debug fax mspi**command for more information.

To display debugging messages for the receiving mail Service Provider Interface (MSPI), use the **debug mspi receive**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mspi receive
no debug mspi receive

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XI	This command was introduced on the Cisco AS5300 universal access server.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.2(4)T	This command was implemented on the Cisco 1750 access router.
12.3(8)T	This command was replaced by the **debug fax mspi**command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug mspi receive**command:

```
Router# debug mspi receive
Jan  1 05:09:33.890: mspi_tel_num_trans: from: Radhika,
ph#in: fax=5271714 ph#dial: 5271714
Jan  1 05:09:33.890:  incoming destPat(5271714), matched(7), tag(22)
Jan  1 05:09:33.890:  out destPat(5......), tag(20), dgt strip enabled
Jan  1 05:09:33.890: mspi_off_new_rcpt: envlp_to [fax=5271714@smith.abccompany.com], 30
Jan  1 05:09:33.890:  tel_numb_dial: 5271714, subaddr:[], cover page
Jan  1 05:09:39.122: mspi_offramp_rfc822_header: msgType=0
Jan  1 05:09:39.122:  envlp_from: [Radhika], 8
Jan  1 05:09:39.122: mspi_off_put_buff: ignore mime type=1, st=CONNECTING, len=0
Jan  1 05:09:39.122: moff_save_buffer: cid=0x1F, mime=9, len=4
Jan  1 05:09:39.122:  offramp disabled receiving!
Dec 31 21:09:44.078: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 5271714
Jan  1 05:09:52.154: mspi_bridge: cid=0x1F, dst cid=0x22, data dir=OFFRAMP, conf dir=DEST
Jan  1 05:09:52.154: mspi_offramp_send_buffer: cid=0x1F, mime=9
Jan  1 05:09:52.154:  buffer with only CR/LF - set buff_len=0
Jan  1 05:09:52.154: mspi_offramp_send_buffer: cid=0x1F, mime=9 rx BUFF_END_OF_PART, offramp
 rcpt enabled
Jan  1 05:09:54.126: mspi_offramp_send_buffer: cid=0x1F, mime=11
Jan  1 05:09:54.134: mspi_offramp_send_buffer: cid=0x1F, mime=11
```

Related Commands

Command	Description
debug mspi send	Displays debugging messages for MSPI send.

debug mspi send

Note Effective with Cisco IOS Release 12.3(8)T, the **debug mspi send**command is replaced by the **debug fax mspi**command. See the **debug fax mspi**command for more information.

To display debugging messages for the sending mail Service Provider Interface (MSPI), use the **debug mspi send**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mspi send
no debug mspi send

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)XI	This command was introduced on the Cisco AS5300 universal access server.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.2(4)T	This command was introduced on the Cisco 1750 access router.
12.3(8)T	This command was replaced by the **debug fax mspi**command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug mspi send** command:

```
Router# debug mspi send
*Oct 16 08:40:27.515: mspi_bridge: cid=0x21, dst cid=0x26, data dir=OFFRAMP, conf
dir=DEST
*Oct 16 08:40:29.143: mspi_setup_req: for cid=0x27
*Oct 16 08:40:29.147:   envelope_from=5??????@fax.cisco.com
*Oct 16 08:40:29.147:   envelope_to=ilyau@cisco.com
*Oct 16 08:40:30.147: mspi_chk_connect: cid=0x27, cnt=0,
*Oct 16 08:40:30.147:  SMTP connected to the server !
*Oct 16 08:40:30.147: mspi_bridge: cid=0x27, dst cid=0x28, data dir=ONRAMP, conf dir=SRC
*Oct 16 08:40:38.995: mspi_xmit: cid=0x27, st=CONFERENCED, src_cid=0x28, buf cnt=0
```

Related Commands

Command	Description
debug mspi receive	Displays debugging messages for MSPI receive.

debug mta receive all

Note Effective with release 12.3(8)T, the **debug mta receiv all**command is replaced by the **debug fax mta**command. See the **debug fax mta**command for more information.

To show output relating to the activity on the Simple Mail Transfer Protocol (SMTP) server, use the **debug mta receive all** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mta receive all
no debug mta receive all

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.2(4)T	This command was implemented on the Cisco 1750 access router.
12.2(8)T	This command was implemented on the Cisco 1751 access routers, Cisco 3725 access routers, and Cisco 3745 access routers.
12.2(13)T	This feature was implemented on the Cisco 7200 series routers.
12.3(8)T	This command was replaced by the **debug fax mta**command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows the messages exchanged (for example, the handshake) between the e-mail server and the off-ramp gateway:

```
Router# debug mta receive all
Jan  1 05:07:41.314: esmtp_server_work: calling helo
Jan  1 05:07:43.354: esmtp_server_work: calling mail
Jan  1 05:07:45.386: esmtp_server_work: calling rcpt
Jan  1 05:07:47.426: esmtp_server_work: calling data
Jan  1 05:07:49.514: (S)R: 'Content-Type: multipart/mixed;
boundary="------------11F7CD9D2EB3E8B8D5627C62"'
Jan  1 05:07:49.514: (S)R: ''
Jan  1 05:07:49.514: esmtp_server_engine_new_part:
Jan  1 05:07:49.514: (S)R: 'Content-Type: text/plain; charset=us-ascii'
Jan  1 05:07:49.514: (S)R: 'Content-Transfer-Encoding: 7bit'
```

```
Jan  1 05:07:49.514: (S)R: ''
Jan  1 05:07:49.514: esmtp_server_engine_new_part:
Jan  1 05:07:49.514: esmtp_server_work: freeing temp header
Jan  1 05:07:49.514: (S)R: 'Content-Type: image/tiff; name="DevTest.8.1610.tif"'
Jan  1 05:07:49.514: (S)R: 'Content-Transfer-Encoding: base64'
Jan  1 05:07:49.514: (S)R: 'Content-Disposition: inline; filename="DevTest.8.1610.tif"'
Jan  1 05:07:49.514: (S)R: ''
Jan  1 05:07:49.514: esmtp_server_engine_update_recipient_status: status=6
Jan  1 05:07:49.514: esmtp_server_engine_new_part:
Jan  1 05:07:49.518: esmtp_server_work: freeing temp header
Jan  1 05:08:03.014: esmtp_server_engine_update_recipient_status: status=7
Jan  1 05:08:04.822: esmtp_server_engine_update_recipient_status: status=6
Jan  1 05:08:33.042: esmtp_server_engine_update_recipient_status: status=7
Jan  1 05:08:34.906: esmtp_server_engine_getline: Unexpected end of file on socket 1
Jan  1 05:08:34.906: esmtp_server_work: error occurred with ctx=0x61FFF710, socket=1
```

Related Commands

Command	Description
debug mta send all	Displays output for all the on-ramp client connections.

debug mta send all

Note Effective with release 12.3(8)T, the **debug mta send all**command is replaced by the **debug fax mta**command. See the **debug fax mta**command for more information.

To display output for all of the on-ramp client connections, use the **debug mta send all** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mta send all
no debug mta send all

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.2(4)T	This command was implemented on the Cisco 1750 access router.
12.2(8)T	This command was implemented on the Cisco 1751 access routers, Cisco 3725 access routers, and Cisco 3745 access routers.
12.2(13)T	This feature was implemented on the Cisco 7200 series routers.
12.3(8)T	This command was replaced by the **debug fax mta** command.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows the messages exchanged (for example, the handshake) between the e-mail server and the on-ramp gateway:

```
Router# debug mta send all
*Oct 16 09:04:13.055: esmtp_client_engine_open: from=5551212@fax.cisco.com,
to=madeup@abccompany.com
*Oct 16 09:04:13.055: esmtp_client_engine_add_headers: from_comment=
*Oct 16 09:04:13.111: esmtp_client_work: socket 0 attempting to connect to IP address
171.71.154.56
*Oct 16 09:04:13.111: esmtp_client_work: socket 0 readable for first time
*Oct 16 09:04:13.135: esmtp_client_work: socket 0 readable for first time
*Oct 16 09:04:13.135: (C)R: 220 madeup.abccompany.com ESMTP Sendmail 8.8.4-Cisco.1/8.6.5
ready at Wed, 27 Sep 2000 11:45:46 -0700 (PDT)
*Oct 16 09:04:13.135: (C)S: EHLO mmoip-c.cisco.com
*Oct 16 09:04:13.183: (C)R: 250-madeup.abccompany.com Hello [172.22.95.16], pleased to meet
 you
*Oct 16 09:04:13.183: (C)R: 250-EXPN
*Oct 16 09:04:13.183: (C)R: 250-VERB
```

Related Commands

Command	Description
debug mta send rcpt-to	Displays output for a specific on-ramp SMTP client connection during an e-mail transmission.

debug mta send rcpt-to

Note Effective with release 12.3(8)T, the **debug mta send rcpt-to**command is no longer available in Cisco IOS.

To display output for a specific on-ramp Simple Mail Transfer Protocol (SMTP) client connection during an e-mail transmission, use the **debug mta send rcpt-to** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mta send rcpt-to *string*
no debug mta send rcpt-to *string*

Syntax Description

string	E-mail address.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)T	This command was introduced.
12.2(4)T	This command was implemented on the Cisco 1750 access router.
12.2(8)T	This command was implemented on the Cisco 1751 access routers, Cisco 3725 access routers, and Cisco 3745 access routers.
12.2(13)T	This feature was implemented on the Cisco 7200 series routers.
12.3(8)T	This command was removed and is no longer available in Cisco IOS software.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows debugging information displayed when the **debug mta send rcpt-to** command has been enabled and the SMTP client is sending an e-mail message:

```
Router# debug mta send rcpt-to 5551212

Router# socket 0 attempting to connect to IP address 100.00.00.00
socket 0 readable for first time - let's try to read it
R:220 madeup.abc.com ESMTP Sendmail 8.8.4-abc.1/8.6.5 ready at Tue, 6
Apr 1999 13:35:39 -0700 (PDT)
S:EHLO mmoip-c.abc.com
R:250-quisp.cisco.com Hello [100.00.00.00], pleased to meet you
R:250-EXPN
R:250-VERB
R:250-8BITMIME
R:250-SIZE
R:250-DSN
```

```
R:250-ETRN
R:250-XUSR
R:250 HELP
S:MAIL FROM:<testing@> RET=HDRS
R:250 <testing@>... Sender ok
S:RCPT TO:<madeup@abc.com> NOTIFY=SUCCESS ORCPT=rfc822;testing@
R:250 <madeup@abc.com>... Recipient ok
R:354 Enter mail, end with "." on a line by itself
S:Received:(Cisco Powered Fax System) by mmoip-c.cisco.com for
<madeup@abc.com> (with Cisco NetWorks); Fri, 17 Oct 1997 14:54:27 +0800
S:To: <madeup@abc.com>
S:Message-ID:<000F1997145427146@mmoip-c.cisco.com>
S:Date:Fri, 17 Oct 1997 14:54:27 +0800
S:Subject:mmoip-c subject here
S:X-Mailer:IOS (tm) 5300 Software (C5300-IS-M)
S:MIME-Version:1.0
S:Content-Type:multipart/mixed;
S: boundary="yradnuoB=_000E1997145426826.mmoip-ccisco.com"
S:From:"Test User" <testing@>
S:--yradnuoB=_000E1997145426826.mmoip-ccisco.com
S:Content-ID:<00101997145427150@mmoip-c.cisco.com>
S:--yradnuoB=_000E1997145426826.mmoip-ccisco.com--
Sending terminating dot ...(socket=0)
S:.
R:250 NAA09092 Message accepted for delivery
S:QUIT
R:221 madeup@abc.com closing connection
Freeing SMTP ctx at 0x6121D454
returned from work_routine, context freed
```

Related Commands

Command	Description
debug mta send all	Displays output for all the on-ramp client connections.

debug mvrp

To display debugging information for Multiple VLAN Registration Protocol (MVRP) configurations, use the **debug mvrp** command in privileged EXEC mode. To disable debugging of MVRP configurations, use the **no** form of this command.

debug mvrp [{**all** | **config** | **error** | **event** | **ha** | **packets** | **switch**}]
no debug mvrp

Syntax Description

all	(Optional) Enables all levels of debugging
config	(Optional) Displays user configuration information.
error	(Optional) Enables error-level debugging.
event	(Optional) Enables event-level debugging.
ha	(Optional) Enables high availability-level debugging.
packets	(Optional) Enables packet-level debugging.
switch	(Optional) Enables switch-level debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXI	This command was introduced.

Usage Guidelines

Conditional interface debugging can be used to limit the scope of output messages related to an interface.

Cisco Catalyst 6000 Series Platforms

On switches with a Switch Processor (SP) or Route Processor (RP), this command can be used only on the SP console.

Examples

The following example shows switch-level debugging enabled:

```
Router# debug mvrp switch
```

Related Commands

Command	Description
clear mvrp statistics	Clears statistics related to MVRP and recorded on one (or all) MVRP-enabled ports.
show mvrp	Displays statistics for configured MVRP attributes on a device or specified ports on a device.

debug mwi relay errors

To debug message waiting indication (MWI) relay errors, use the **debug mwi relay errors**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mwi relay errors
no debug mwi relay errors

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XT	This command was introduced on the following platforms: Cisco 1750, Cisco 1751, Cisco 2600 series and Cisco 3600 series multiservice routers; and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug mwi relay errors** command provides a debug monitor display of any error messages, when MWI Relay Server (Cisco IOS Telephony Server) is trying to do MWI Relay to extensions on remote Cisco IOS Telephony Service (ITS).

Examples

The following examples show errors when MWI Relay Server tries to do an MWI Relay to extension 7004, but location of 7004 is not known to the MWI Relay Server:

```
Router#
debug mwi relay errors

mwi-relay error info debugging is on
01:46:48: MWI-APP: mwi_notify_status: No ClientID (7004) registered
```

Related Commands

Command	Description
debug ephone mwi	Sets MWI debugging for the Cisco IOS Telephony Service router.
debug mwi relay events	Sets MWI relay events debugging for the Cisco IOS Telephony Service router.

debug mwi relay events

To set message waiting indication (MWI) relay events debugging, use the **debug mwi relay events**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mwi relay events
no debug mwi relay events

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XT	This command was introduced on the following platforms: Cisco 1750, Cisco 1751, Cisco 2600 series and Cisco 3600 series multiservice routers; and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug mwi relay events** command provides a debug monitor display of events, when MWI Relay Server (Cisco IOS Telephony Server) is trying to do MWI Relay to extensions on remote Cisco IOS Telephony Services (ITS).

Examples

The following debugging messages are shown when the MWI Relay server tries to send MWI Information to remote client 7001 and the location of 7001 is known by the MWI Relay Server:

```
Router# debug mwi relay events

mwi-relay events info debugging is on
01:45:34: mwi_notify_status: Queued event for mwi_app_queue
01:45:34: MWI-APP: mwi_app_process_event:
01:45:34: MWI-APP: mwi_app_process_event: MWI Event for ClientID(7001)@(1.8.17.22)
```

Related Commands

Command	Description
debug ephone mwi	Sets MWI debugging for the Cisco IOS Telephony Service router.
debug mwi relay errors	Sets MWI relay errors debugging for the Cisco IOS Telephony Service router.

debug ncia circuit through debug pxf tbridge

debug nat64

To enable stateless Network Address Translation 64 (NAT64) debugging, use the **debug nat64** command in privileged EXEC mode. To disable NAT64 debugging, use the **no** form of this command.

debug nat64 {**all** | {**aliases** | **ha** {**all** | **info** | **trace** | **warn**}} | **id-manager** | **info** | **intf-address** | **issu** {**all** | **message** | **trace**} | **memory** | **pool-routes** | **statistics** | **trace** | **warn**}
no debug nat64 {**all** | {**aliases** | **ha** {**all** | **info** | **trace** | **warn**}} | **id-manager** | **info** | **intf-address** | **issu** {**all** | **message** | **trace**} | **memory** | **pool-routes** | **statistics** | **trace** | **warn**}

Syntax Description

all	Enables information, trace, and warning level debugging.
aliases	Enables debugging of IP aliases created by NAT64.
ha	Enables high availability (HA) debugging.
all	Enables HA information, trace, and warning level debugging.
info	Enables HA information level debugging.
trace	Enables HA trace level debugging.
warn	Enables HA warning level debugging.
id-manager	Enables Interface Descriptor manager trace debugging.
info	Enables information level debugging.
intf-address	Enables interface address change events debugging.
issu	Enables In-Service Software Upgrade (ISSU) debugging.
all	Enables ISSU trace level and message debugging.
message	Enables ISSU message debugging.
trace	Enables ISSU trace level debugging.
memory	Enables memory trace debugging.
pool-routes	Enables the debugging of routes attached to a a pool address range.
statistics	Enables statistics debugging.
trace	Enables trace level debugging.
warn	Enables warning level debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was modified. The **aliases**, **intf-address**, and **pool-routes** keywords were added.
15.4(1)T	This command was integrated into Cisco IOS Release 15.4(1)T.

Usage Guidelines

The general debugging levels are information, trace, and warning. The **debug nat64 memory** and **debug nat64 id-manager** commands provide detailed traces related to resources and memory allocation. The **debug nat64 issu** command provides traces specific to the ISSU messages exchanged.

Examples

The following is sample output from the **debug nat64 statistics** command. The output fields are self-explanatory.

```
Router# debug nat64 statistics

NAT64 statistics debugging is on
Sep 16 18:26:24.537 IST: NAT64 (stats): Received stats update for IDB(FastEthernet0/3/5)
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v4v6 from 94368894 to
95856998 (is_delta(TRUE) value(1488104))
Sep 16 18:26:24.537 IST: NAT64 (stats): Received stats update for IDB(FastEthernet0/3/4)
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v6v4 from 7771538 to 7894088
 (is_delta(TRUE) value(122550))
Sep 16 18:26:24.537 IST: NAT64 (stats): Received global stats update
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v4v6 from 1718650332 to
1720138437 (is_delta(TRUE) value(1488105))
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v6v4 from 1604459283 to
1604581833 (is_delta(TRUE) value(122550))
```

The following is sample output from the **debug nat64 memory** command. The output fields are self-explanatory.

```
Router# debug nat64 memory

NAT64 memory debugging is on
Sep 16 18:28:03.713 IST: NAT64 (memory): Allocated 0x7FFA7DA2F750
Sep 16 18:28:03.713 IST: NAT64 (memory): Allocated 0x7FFA9EC00D30
Sep 16 18:28:03.713 IST: NAT64 (memory): Allocated 0x7FFA9D1532C8
```

Related Commands

Command	Description
nat64 enable	Enables stateless NAT64 on an interface.

debug ncia circuit

To display circuit-related information between the native client interface architecture (NCIA) server and client, use the **debugncia circuit**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ncia circuit [{**error** | **event** | **flow-control** | **state**}]
no debug ncia circuit [{**error** | **event** | **flow-control** | **state**}]

Syntax Description

error	(Optional) Displays the error situation for each circuit.
event	(Optional) Displays the packets received and sent for each circuit.
flow-control	(Optional) Displays the flow control information for each circuit.
state	(Optional) Displays the state changes for each circuit.

Command Modes

Privileged EXEC

Usage Guidelines

NCIA is an architecture developed by Cisco for accessing Systems Network Architecture (SNA) applications. This architecture allows native SNA interfaces on hosts and clients to access TCP/IP backbones.

You cannot enable debugging output for a particular client or particular circuit.

Caution Do not enable the **debugncia circuit** command during normal operation because this command generates a substantial amount of output messages and could slow down the router.

Examples

The following is sample output from the**debugncia circuit error** command. In this example, the possible errors are displayed. The first error message indicates that the router is out of memory. The second message indicates that the router has an invalid circuit control block. The third message indicates that the router is out of memory. The remaining messages identify errors related to the finite state machine.

```
Router# debug ncia circuit error
NCIA: ncia_circuit_create memory allocation fail
NCIA: ncia_send_ndlc: invalid circuit control block
NCIA: send_ndlc: fail to get buffer for ndlc primitive xxx
NCIA: ncia circuit fsm: Invalid input
NCIA: ncia circuit fsm: Illegal state
NCIA: ncia circuit fsm: Illegal input
NCIA: ncia circuit fsm: Unexpected input
NCIA: ncia circuit fsm: Unknown error rtn code
```

The following is sample output from the **debugncia circuit event**command. In this example, a session startup sequence is displayed.

```
Router# debug ncia circuit event
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_START_DL, Len: 24, tmac: 4000.1060.1000,
          tsap: 4, csap 8, oid: 8A91E8, tid 0, lfs 16, ws 1
NCIA: create circuit: saddr 4000.1060.1000, ssap 4, daddr 4000.3000.0003, dsap 8 sid:
```

```
          8B09A8
NCIA: send NDLC_DL_STARTED to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_DL_STARTED, Len: 2,4 tmac: 4000.1060.1000,
          tsap: 4, csap 8, oid: 8A91E8, tid 8B09A8, lfs 16, ws 1
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_XID_FRAME, Len: 12, sid: 8B09A8, FC 0x81
NCIA: send NDLC_XID_FRAME to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_XID_FRAME, Len: 12, sid: 8A91E8, FC 0xC1
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_XID_FRAME, Len: 18, sid: 8B09A8, FC 0xC1
NCIA: send NDLC_CONTACT_STN to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_CONTACT_STN, Len: 12, sid: 8A91E8, FC 0xC1
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_STN_CONTACTED, Len: 12, sid: 8B09A8, FC 0xC1
NCIA: send NDLC_INFO_FRAME to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_INFO_FRAME, Len: 30, sid: 8A91E8, FC 0xC1
```

The following table describes the significant fields shown in the display.

Table 34: debug ncia circuit event Field Descriptions

Field	Description
IN	Incoming message from client.
OUT	Outgoing message to client.
Ver_Id	NDLC version ID.
MsgType	NDLC message type.
Len	NDLC message length.
tmac	Target MAC.
tsap	Target SAP.
csap	Client SAP.
oid	Origin ID.
tid	Target ID.
lfs	Largest frame size flag.
ws	Window size.
saddr	Source MAC address.
ssap	Source SAP.
daddr	Destination MAC address.
dsap	Destination SAP.
sid	Session ID.
FC	Flow control flag.

In the following messages, an NDLC_START_DL messages is received from a client to start a data-link session:

```
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_START_DL, Len: 24, tmac: 4000.1060.1000,
          tsap: 4, csap 8, oid: 8A91E8, tid 0, lfs 16, ws 1
NCIA: create circuit: saddr 4000.1060.1000, ssap 4, daddr 4000.3000.0003, dsap 8 sid:
          8B09A8
```

The next two messages indicate that an NDLC_DL_STARTED message is sent to a client. The server informs the client that a data-the link session is started.

```
NCIA: send NDLC_DL_STARTED to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_DL_STARTED, Len: 2,4 tmac: 4000.1060.1000,
          tsap: 4, csap 8, oid: 8A91E8, tid 8B09A8, lfs 16, ws 1
```

In the following two messages, an NDLC_XID_FRAME message is received from a client, and the client starts an XID exchange:

```
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_XID_FRAME, Len: 12, sid: 8B09A8, FC 0x81
NCIA: send NDLC_XID_FRAME to client 10.2.20.3 for ckt: 8B09A8
```

In the following two messages, an NDLC_XID_FRAME message is sent from a client, and an DLC_XID_FRAME message is received from a client:

```
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_XID_FRAME, Len: 12, sid: 8A91E8, FC 0xC1
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_XID_FRAME, Len: 18, sid: 8B09A8, FC 0xC1
```

The next two messages show that an NDLC_CONTACT_STN message is sent to a client:

```
NCIA: send NDLC_CONTACT_STN to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_CONTACT_STN, Len: 12, sid: 8A91E8, FC 0xC1
```

In the following message, an NDLC_STN_CONTACTED message is received from a client. The client informs the server that the station has been contacted.

```
NCIA(IN): Ver_Id: 0x81, MsgType: NDLC_STN_CONTACTED, Len: 12, sid: 8B09A8, FC 0xC1
```

In the last two messages, an NDLC_INFO_FRAME is sent to a client, and the server sends data to the client:

```
NCIA: send NDLC_INFO_FRAME to client 10.2.20.3 for ckt: 8B09A8
NCIA(OUT): Ver_Id: 0x81, MsgType: NDLC_INFO_FRAME, Len: 30, sid: 8A91E8, FC 0xC1
```

The following is sample output from the **debugnciacircuitflow-control** command. In this example, the flow control in a session startup sequence is displayed:

```
Router# debug ncia circuit flow-control
NCIA: no flow control in NDLC_DL_STARTED frame
NCIA: receive Increment Window Op for circuit 8ADE00
NCIA: ncia_flow_control_in FC 0x81, IW 1 GP 2 CW 2, Client IW 1 GP 0 CW 1
NCIA: grant client more packet by sending Repeat Window Op
NCIA: ncia_flow_control_out FC: 0xC1, IW 1 GP 2 CW 2, Client IW 1 GP 2 CW 2
NCIA: receive FCA for circuit 8ADE00
NCIA: receive Increment Window Op for circuit 8ADE00
NCIA: ncia_flow_control_in FC 0xC1, IW 1 GP 5 CW 3, Client IW 1 GP 2 CW 2
NCIA: grant client more packet by sending Repeat Window Op
NCIA: ncia_flow_control_out FC: 0xC1, IW 1 GP 5 CW 3, Client IW 1 GP 5 CW 3
NCIA: receive FCA for circuit 8ADE00
NCIA: receive Increment Window Op for circuit 8ADE00
NCIA: ncia_flow_control_in FC 0xC1, IW 1 GP 9 CW 4, Client IW 1 GP 5 CW 3
NCIA: grant client more packet by sending Repeat Window Op
```

```
NCIA: ncia_flow_control_out FC: 0xC1, IW 1 GP 8 CW 4, Client IW 1 GP 9 CW 4
NCIA: reduce ClientGrantPacket by 1 (Granted: 8)
NCIA: receive FCA for circuit 8ADE00
NCIA: receive Increment Window Op for circuit 8ADE00
```

The following table describes the significant fields shown in the display.

Table 35: debug ncia circuit flow-control Field Descriptions

Field	Description
IW	Initial window size.
GP	Granted packet number.
CW	Current window size.

The following is sample output from the **debugnciacircuitstate**command. In this example, a session startup sequence is displayed:

```
Router# debug ncia circuit state
NCIA: pre-server fsm: event CONN_OPENED
NCIA: pre-server fsm: event NDLC_PRIMITIVES
NCIA: server event: WAN - STDL state: CLSOED
NCIA: ncia server fsm action 32
NCIA: circuit state: CLOSED -> START_DL_RCVD
NCIA: server event: DLU - TestStn.Rsp state: START_DL_RCVD
NCIA: ncia server fsm action 17
NCIA: circuit state: START_DL_RCVD -> DL_STARTED_SND
NCIA: pre-server fsm: event NDLC_PRIMITIVES
NCIA: server event: WAN - XID state: DL_STARTED_SND
NCIA: ncia server fsm action 33
NCIA: circuit state: DL_STARTED_SND -> DL_STARTED_SND
NCIA: server event: DLU - ReqOpnStn.Req state: DL_STARTED_SND
NCIA: ncia server fsm action 33
NCIA: circuit state: DL_STARTED_SND -> OPENED
NCIA: server event: DLU - Id.Rsp state: OPENED
NCIA: ncia server fsm action 11
NCIA: circuit state: OPENED -> OPENED
NCIA: pre-server fsm: event NDLC_PRIMITIVES
NCIA: server event: WAN - XID state: OPENED
NCIA: ncia server fsm action 33
NCIA: circuit state: OPENED -> OPENED
NCIA: server event: DLU - Connect.Req state: OPENED
NCIA: ncia server fsm action 6
NCIA: circuit state: OPENED -> CONNECT_PENDING
NCIA: pre-server fsm: event NDLC_PRIMITIVES
NCIA: server event: WAN - CONR state: CONNECT_PENDING
NCIA: ncia server fsm action 33 --> CLS_CONNECT_CNF sets NciaClsBusy
NCIA: circuit state: CONNECT_PENDING -> CONNECTED
NCIA: server event: DLU - Flow.Req (START) state: CONNECTED
NCIA: ncia server fsm action 25 --> unset NciaClsBusy
NCIA: circuit state: CONNECTED -> CONNECTED
NCIA: server event: DLU - Data.Rsp state: CONNECTED
NCIA: ncia server fsm action 8
NCIA: circuit state: CONNECTED -> CONNECTED
```

The following table describes the significant fields shown in the display.

Table 36: debug ncia circuit state Field Descriptions

Field	Description
WAN	Event from WAN (client).
DLU	Event from upstream module--dependent logical unit (DLU).
ADMIN	Administrative event.
TIMER	Timer event.

Related Commands

Command	Description
debug dmsp fax-to-doc	Enables debugging of DLSw+.
debug ncia client	Displays debug information for all NCIA client processing that occurs in the router.
debug ncia server	Displays debug information for the NCIA server and its upstream software modules.

debug ncia client

To display debug information for all native client interface architecture (NCIA) client processing that occurs in the router, use the **debug ncia client**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ncia client [{*ip-address* | **error** [*ip-address*] | **event** [*ip-address*] | **message** [*ip-address*]}]
no debug ncia client [{*ip-address* | **error** [*ip-address*] | **event** [*ip-address*] | **message** [*ip-address*]}]

Syntax Description

ip-address	(Optional) The remote client IP address.
error	(Optional) Triggers the recording of messages only when errors occur. The current state and event of an NCIA client are normally included in the message. If you do not specify an IP address, the error messages are logged for all active clients.
event	(Optional) Triggers the recording of messages that describe the current state and event--and sometimes the action that just completed--for the NCIA client. If you do not specify an IP address, the messages are logged for all active clients.
message	(Optional) Triggers the recording of messages that contain up to the first 32 bytes of data in a TCP packet sent to or received from an NCIA client. If you do not specify an IP address, the messages are logged for all active clients.

Command Modes

Privileged EXEC

Usage Guidelines

NCIA is an architecture developed by Cisco for accessing Systems Network Architecture (SNA) applications. This architecture allows native SNA interfaces on hosts and clients to access TCP/IP backbones.

Use the **debug ncia client error** command to see only certain error conditions that occur.

Use the **debug ncia client event** command to determine the sequences of activities that occur while an NCIA client is in different processing states.

Use the **debug ncia client message** command to see only the first 32 bytes of data in a TCP packet sent to or received from an NCIA client.

The **debug ncia client**command can be used in conjunction with the **debug ncia server** and **debug ncia circuit** commands to get a complete picture of NCIA activity.

Examples

The following is sample output from the **debug ncia client** command. Following the example is a description of each sample output message.

```
Router# debug ncia client
NCIA: Passive open 10.2.20.123(1088) -> 1973
NCIA: index for client hash queue is 27
NCIA: number of element in client hash queue 27 is 1
NCIA: event PASSIVE_OPEN, state NCIA_CLOSED for client 10.2.20.123
NCIA: Rcvd msg type NDLC_CAP_XCHG in tcp packet for client 10.2.20.123
NCIA: First 17 byte of data rcvd: 8112001100000000000000400050104080C
NCIA: Sent msg type NDLC_CAP_XCHG in tcp packet to client 10.2.20.123
NCIA: First 17 byte of data sent: 81120011100000001000040005010408OC
NCIA: event CAP_CMD_RCVD, state NCIA_CAP_WAIT, for client 10.2.20.123, cap xchg cmd sent
NCIA: Rcvd msg type NDLC_CAP_XCHG in tcp packet for client 10.2.20.123
```

```
NCIA: First 17 byte of data rcvd: 81120011100000001000000050104080C
NCIA: event CAP_RSP_RCVD, state NCIA_CAP_NEG for client 10.2.20.123
NCIA: Rcvd msg type NDLC_PEER_TEST_REQ in tcp packet for client 10.2.20.123
NCIA: First 4 byte of data rcvd: 811D0004
NCIA: event KEEPALIVE_RCVD, state NCIA_OPENED for client 10.2.20.123
NCIA: Sent msg type NDLC_PEER_TEST_RSP in tcp packet to client 10.2.20.123
NCIA: First 4 byte of data sent: 811E0004IA
NCIA: event TIME_OUT, state NCIA_OPENED, for client 10.2.20.123, keepalive_count = 0
NCIA: Sent msg type NDLC_PEER_TEST_REQ, in tcp packet to client 10.2.20.123
NCIA: First 4 byte of data sent: 811D0004
NCIA: Rcvd msg type NDLC_PEER_TEST_RSP in tcp packet for client 10.2.20.123
NCIA: First 4 byte of data rcvd: 811E0004
NCIA: event KEEPALIVE_RSP_RCVD, state NCIA_OPENED for client 10.2.20.123
NCIA: Error, event PASIVE_OPEN, state NCIA_OPENED, for client 10.2.20.123, should not have
 occurred.
NCIA: Error, active_open for pre_client_fsm while client 10.2.20.123 is active or not
configured, registered.
```

Messages in lines 1 through 12 show the events that occur when a client connects to the router (the NCIA server). These messages show a passive_open process.

Messages in lines 13 to 17 show the events that occur when a TIME_OUT event is detected by a client PC workstation. The workstation sends an NDLC_PEER_TEST_REQ message to the NCIA server, and the router responds with an NDLC_PEER_TEST_RSP message.

Messages in lines 18 to 23 show the events that occur when a TIME_OUT event is detected by the router (the NCIA server). The router sends an NDLC_PEER_TEST_REQ message to the client PC workstation, and the PC responds with an NDLC_PEER_TEST_RSP message.

When you use the **debug ncia client message** command, the messages shown on lines 6, 8, 11, 14, 17, 20, and 22 are output in addition to other messages not shown in this example.

When you use the **debug ncia client error** command, the messages shown on lines 24 and 25 are output in addition to other messages not shown in this example.

Related Commands

Command	Description
debug ncia circuit	Displays debug information for all NCIA client processing that occurs in the router.
debug ncia server	Displays debug information for the NCIA server and its upstream software modules.

debug ncia server

To display debug information for the native client interface architecture (NCIA) server and its upstream software modules, use the **debug ncia server**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ncia server
no debug ncia server

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

NCIA is an architecture developed by Cisco for accessing Systems Network Architecture (SNA) applications. This architecture allows native SNA interfaces on hosts and clients to access TCP/IP backbones.

The **debug ncia server** command displays all Cisco Link Services (CLS) messages between the NCIA server and its upstream modules, such as data-link switching (DLSw) and downstream physical units (DSPUs). Use this command when a problem exists between the NCIA server and other software modules within the router.

You cannot enable debugging output for a particular client or particular circuit.

Examples

The following is sample output from the **debug ncia server** command. In this example, a session startup sequence is displayed. Following the example is a description of each group of sample output messages.

```
Router# debug ncia server
NCIA: send CLS_TEST_STN_IND to DLU
NCIA: Receive TestStn.Rsp
NCIA: send CLS_ID_STN_IND to DLU
NCIA: Receive ReqOpnStn.Req
NCIA: send CLS_REQ_OPNSTN_CNF to DLU
NCIA: Receive Id.Rsp
NCIA: send CLS_ID_IND to DLU
NCIA: Receive Connect.Req
NCIA: send CLS_CONNECT_CNF to DLU
NCIA: Receive Flow.Req
NCIA: Receive Data.Req
NCIA: send CLS_DATA_IND to DLU
NCIA: send CLS_DISC_IND to DLU
NCIA: Receive Disconnect.Rsp
```

In the following messages, the client is sending a test message to the host and the test message is received by the host:

```
NCIA: send CLS_TEST_STN_IND to DLU
NCIA: Receive TestStn.Rsp
```

In the next message, the server is sending an exchange identification (XID) message to the host:

```
NCIA: send CLS_ID_STN_IND to DLU
```

In the next two messages, the host opens the station and the server responds:

```
NCIA: Receive ReqOpnStn.Req
NCIA: send CLS_REQ_OPNSTN_CNF to DLU
```

In the following two messages, the client is performing an XID exchange with the host:

```
NCIA: Receive Id.Rsp
NCIA: send CLS_ID_IND to DLU
```

In the next group of messages, the host attempts to establish a session with the client:

```
NCIA: Receive Connect.Req
NCIA: send CLS_CONNECT_CNF to DLU
NCIA: Receive Flow.Req
```

In the next two messages, the host sends data to the client:

```
NCIA: Receive Data.Req
NCIA: send CLS_DATA_IND to DLU
```

In the last two messages, the client closes the session:

```
NCIA: send CLS_DISC_IND to DLU
NCIA: Receive Disconnect.Rsp
```

Related Commands

Command	Description
debug dmsp fax-to-doc	Enables debugging of DLSw+.
debig mcoa circuit	Displays circuit-related information between the NCIA server and client.
debug ncia client	Displays debug information for all NCIA client processing that occurs in the router.

debug netbios error

To display information about Network Basic Input/Output System (NetBIOS) protocol errors, use the **debug netbios error**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug netbios error
no debug netbios error

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

For complete information on the NetBIOS process, use the **debug netbios packet** command along with the **debug netbios error** command.

Examples

The following is sample output from the **debug netbios error**command. This example shows that an illegal packet has been received on the asynchronous interface.

```
Router# debug netbios error
Async1 nbf Bad packet
```

Related Commands

Command	Description
debug netbios-name-cache	Displays name caching activities on a router.
debug netbios packet	Displays general information about NetBIOS packets.

debug netbios packet

To display general information about Network Basic Input/Output System (NetBIOS) packets, use the **debug netbios packet**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug netbios packet
no debug netbios packet

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

For complete information on the NetBIOS process, use the **debug netbios error** command along with the **debug netbios packet** command.

Examples

The following is sample output from the **debug netbios packet**and **debug netbios error** commands. This example shows the Logical Link Control (LLC) header for an asynchronous interface followed by the NetBIOS information. For additional information on the NetBIOS fields, refer to *IBM LAN Technical Reference IEEE 802.2* .

```
Router# debug netbios packet
Async1 (i) U-format UI C_R=0x0
(i) NETBIOS_ADD_NAME_QUERY
 Resp_correlator= 0x6F 0x0
 Src name=CS-NT-1
Async1 (i) U-format UI C_R=0x0
(i) NETBIOS_ADD_GROUP_QUERY
 Resp_correlator= 0x6F 0x0
 Src name=COMMSERVER-WG
Async1 (i) U-format UI C_R=0x0
(i) NETBIOS_ADD_NAME_QUERY
 Resp_correlator= 0x6F 0x0
 Src name=CS-NT-1
Ethernet0 (i) U-format UI C_R=0x0
(i) NETBIOS_DATAGRAM
 Length= 0x2C 0x0
 Dest name=COMMSERVER-WG
 Src name=CS-NT-3
```

Related Commands

Command	Description
debug netbios error	Displays information about NetBIOS protocol errors.
debug netbios-name-cache	Displays name caching activities on a router.

debug netbios-name-cache

To display name caching activities on a router, use the **debugnetbios-name-cache**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug netbios-name-cache
no debug netbios-name-cache

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Examine the display to diagnose problems in Network Basic Input/Output System (NetBIOS) name caching.

Examples

The following is sample output from the **debugnetbios-name-cache**command:

```
Router# debug netbios-name-cache
NETBIOS: L checking name ORINDA, vrn=0
NetBIOS name cache table corrupted at offset 13
NetBIOS name cache table corrupted at later offset, at location 13
NETBIOS: U chk name=ORINDA, addr=1000.4444.5555, idb=TR1, vrn=0, type=1
NETBIOS: U upd name=ORINDA,addr=1000.4444.5555,idb=TR1,vrn=0,type=1
NETBIOS: U add name=ORINDA,addr=1000.4444.5555,idb=TR1,vrn=0,type=1
NETBIOS: U no memory to add cache entry. name=ORINDA,addr=1000.4444.5555
NETBIOS: Invalid structure detected in netbios_name_cache_ager
NETBIOS: flushed name=ORINDA, addr=1000.4444.5555
NETBIOS: expired name=ORINDA, addr=1000.4444.5555
NETBIOS: removing entry. name=ORINDA,addr=1000.4444.5555,idb=TR1,vrn=0
NETBIOS: Tossing ADD_NAME/STATUS/NAME/ADD_GROUP frame
NETBIOS: Lookup Failed -- not in cache
NETBIOS: Lookup Worked, but split horizon failed
NETBIOS: Could not find RIF entry
NETBIOS: Cannot duplicate packet in netbios_name_cache_proxy
```

Note The sample display is a composite output. Debugging output that you actually see would not necessarily occur in this sequence.

The following table describes the significant fields shown in the display.

Table 37: debug netbios-name-cache Field Descriptions

Field	Description
NETBIOS	NetBIOS name caching debugging output.
L, U	L means lookup; U means update.
addr=1000.4444.5555	MAC address of machine being looked up in NetBIOS name cache.
idb=TR1	Indicates that the name of machine was learned from Token Ring interface number 1; idb is into interface data block.

Field	Description
vrn=0	Packet comes from virtual ring number 0. This packet actually comes from a real Token Ring interface, because virtual ring number 0 is not valid.
type=1	Indicates the way that the router learned about the specified machine. The possible values are as follows: • 1--Learned from traffic • 2--Learned from a remote peer • 4--Statically entered via the configuration of the router

With the first line of output, the router declares that it has examined the NetBIOS name cache table for the machine name ORINDA and that the packet that prompted the lookup came from virtual ring 0. In this case, this packet comes from a real interface--virtual ring number 0 is not valid.

```
NETBIOS: L checking name ORINDA, vrn=0
```

The following two lines indicate that an invalid NetBIOS entry exists and that the corrupted memory was detected. The invalid memory will be removed from the table; no action is needed.

```
NetBIOS name cache table corrupted at offset 13
NetBIOS name cache table corrupted at later offset, at location 13
```

The following line indicates that the router attempted to check the NetBIOS cache table for the name ORINDA with MAC address 1000.4444.5555. This name was obtained from Token Ring interface 1. The type field indicates that the name was learned from traffic.

```
NETBIOS: U chk name=ORINDA, addr=1000.4444.5555, idb=TR1, vrn=0, type=1
```

The following line indicates that the NetBIOS name ORINDA is in the name cache table and was updated to the current value:

```
NETBIOS: U upd name=ORINDA,addr=1000.4444.5555,idb=TR1,vrn=0,type=1
```

The following line indicates that the NetBIOS name ORINDA is not in the table and must be added to the table:

```
NETBIOS: U add name=ORINDA,addr=1000.4444.5555,idb=TR1,vrn=0,type=1
```

The following line indicates that there was insufficient cache buffer space when the router tried to add this name:

```
NETBIOS: U no memory to add cache entry. name=ORINDA,addr=1000.4444.5555
```

The following line indicates that the NetBIOS ager detects an invalid memory in the cache. The router clears the entry; no action is needed.

```
NETBIOS: Invalid structure detected in netbios_name_cache_ager
```

The following line indicates that the entry for ORINDA was flushed from the cache table:

```
NETBIOS: flushed name=ORINDA, addr=1000.4444.5555
```

The following line indicates that the entry for ORINDA timed out and was flushed from the cache table:

```
NETBIOS: expired name=ORINDA, addr=1000.4444.5555
```

The following line indicates that the router removed the ORINDA entry from its cache table:

```
NETBIOS: removing entry. name=ORINDA,addr=1000.4444.5555,idb=TR1,vrn=0
```

The following line indicates that the router discarded a NetBIOS packet of type ADD_NAME, STATUS, NAME_QUERY, or ADD_GROUP. These packets are discarded when multiple copies of one of these packet types are detected during a certain period of time.

```
NETBIOS: Tossing ADD_NAME/STATUS/NAME/ADD_GROUP frame
```

The following line indicates that the system could not find a NetBIOS name in the cache:

```
NETBIOS: Lookup Failed -- not in cache
```

The following line indicates that the system found the destination NetBIOS name in the cache, but located on the same ring from which the packet came. The router will drop this packet because the packet should not leave this ring.

```
NETBIOS: Lookup Worked, but split horizon failed
```

The following line indicates that the system found the NetBIOS name in the cache, but the router could not find the corresponding RIF. The packet will be sent as a broadcast frame.

```
NETBIOS: Could not find RIF entry
```

The following line indicates that no buffer was available to create a NetBIOS name cache proxy. A proxy will not be created for the packet, which will be forwarded as a broadcast frame.

```
NETBIOS: Cannot duplicate packet in netbios_name_cache_proxy
```

Related Commands

Command	Description
debug netbios error	Displays information about NetBIOS protocol errors.
debug netbios packet	Displays general information about NetBIOS packets.

debug netconf

To enable debugging of network configuration protocol (NETCONF) sessions, use the **debug netconf** command in privileged EXEC mode. To turn off NETCONF debugging, use the **no** form of this command.

debug netconf {**all** | **error**}
no debug netconf {**all** | **error**}

Syntax Description

all	Enables debugging of NETCONF sessions, including NETCONF errors.
error	Enables debugging of NETCONF errors.

Command Default

NETCONF debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

The **debug netconf**command issues debug information only when an operational error has happened. In most situations, the NETCONF notifications sent between the NETCONF Network Manager and the client are sufficient to diagnose most NETCONF problems.

To view Extensible Markup Language (XML) parsing errors when using NETCONF over SSHv2, you must also configure the **debug cns xml all** command.

Examples

The following example shows how to enable debugging of all NETCONF sessions:

```
Router# debug netconf

00:14:03: NETCONF-ERROR: could not find user1
00:14:03: NETCONF-ERROR: could not find tftp://samplelocation/samplefile
00:14:03: NETCONF: locking 1 by session 646B7038
00:14:03: NETCONF: locking 2 by session 646B7038
00:14:03: NETCONF: locking 1 by session 646B7038
00:14:03: NETCONF-ERROR: invalid session unlock attempt
00:14:03: NETCONF: locking 1 by session 646B7038
00:14:03: NETCONF-ERROR: lock already active
00:14:13: NETCONF-ERROR: lock time 1 expired closing session 646B7038
```

The following table describes the significant fields shown in the display.

Table 38: debug netconf Field Descriptions

Field	Description
NETCONF-ERROR: could not find user1	NETCONF could not find the specified username.
NETCONF-ERROR: could not find tftp://samplelocation/samplefile	NETCONF could not find the specified file path.
NETCONF: locking 1 by session 646B7038	This user is locking NETCONF.
NETCONF-ERROR: invalid session unlock attempt	Another user is trying to unlock NETCONF without first acquiring the lock.
NETCONF-ERROR: lock already active	Another user is trying to lock NETCONF while it is currently locked.
NETCONF-ERROR: lock time 1 expired closing session 646B7038	A locked NETCONF session has been idle longer than the time configured by the **netconf lock-time** command. The locked NETCONF session is closed.

Related Commands

Command	Description
clear netconf	Clears NETCONF statistics counters, NETCONF sessions, and frees associated resources and locks.
debug cns xml	Turns on debugging messages related to the CNS XML parser.
netconf lock-time	Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation.
netconf max-sessions	Specifies the maximum number of concurrent NETCONF sessions allowed.
netconf ssh	Enables NETCONF over SSHv2.
show netconf	Displays NETCONF statistics counters and session information.

debug nextport vsmgr detail

To turn on debugging for NextPort voice services, use the **debug nextport vsmgr detail**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug nextport vsmgr detail
no debug nextport vsmgr detail

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(14)T	T.38 fax relay call statistics were made available to Call Detail Records (CDRs) through Vendor-Specific Attributes (VSAs) and added to the call log.

Usage Guidelines

This command debugs digital signal processor (DSP) message exchanges between applications and the DSP.

Examples

The following examples turn on debugging for NextPort voice services:

debug nextport vsmgr detail Command on the Originating Gateway

```
Router# debug nextport vsmgr detail
NextPort Voice Service Manager:
  NP Voice Service Manager Detail debugging is on
.
.
.
May  7 21:09:49.135 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:49.195 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:49.291 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:09:51.191 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:51.331 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:09:51.803 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:09:51.803 UTC: request_id = 0x01, request_type = 0x0F
May  7 21:09:51.803 UTC: VOICE_TRANSMIT_STATS(1/2): num_voice_packets 4 num_sig_packets 0
num_cn_packets 1 transmit_duration 8FC end_point_detection 0
May  7 21:09:51.803 UTC: VOICE_RECEIVE_STATS(1/2): num_voice_packets 4 num_sig_packets 0
num_cn_packets 1 receive_duration 8FC  voice_receive_duration 0 num_pos_packets 0
num_bph_packets 0 num_late_packets 0 num_early_packets 0
May  7 21:09:51.803 UTC: VOICE_PLAYOUT_DELAY_STATS(1/2): curr_playout_delay 0
min_playout_delay 0 max_playout_delay 0 clock offset 0
May  7 21:09:51.803 UTC: VOICE_PLAYOUT_ERROR(1/2): pred_conceal 0x0 inter_conceal 0x0
silence_conceal 0x0 buffer_overflow 0x0 endpt_det_error 0x0
```

```
May  7 21:09:53.231 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:09:53.231 UTC: request_id = 0x01, request_type = 0x0F
May  7 21:09:53.231 UTC: VOICE_TRANSMIT_STATS(1/2): num_voice_packets 1E num_sig_packets 0
 num_cn_packets 1 transmit_duration E92 end_point_detection 0
May  7 21:09:53.231 UTC: VOICE_RECEIVE_STATS(1/2): num_voice_packets 4 num_sig_packets 0
num_cn_packets 1 receive_duration E92  voice_receive_duration 0 num_pos_packets 0
num_bph_packets 0 num_late_packets 0 num_early_packets 0
May  7 21:09:53.231 UTC: VOICE_PLAYOUT_DELAY_STATS(1/2): curr_playout_delay 5A
min_playout_delay 5A max_playout_delay 5A clock offset 19778906
May  7 21:09:53.231 UTC: VOICE_PLAYOUT_ERROR(1/2): pred_conceal 0x0 inter_conceal 0x0
silence_conceal 0x0 buffer_overflow 0x0 endpt_det_error 0x0
May  7 21:09:56.055 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:09:56.055 UTC: request_id = 0x01, request_type = 0x0F
May  7 21:09:56.055 UTC: VOICE_TRANSMIT_STATS(1/2): num_voice_packets 23 num_sig_packets 0
 num_cn_packets 2 transmit_duration 19A0 end_point_detection BB8
May  7 21:09:56.055 UTC: VOICE_RECEIVE_STATS(1/2): num_voice_packets 8A num_sig_packets 0
num_cn_packets 1 receive_duration 19A0  voice_receive_duration 0 num_pos_packets 0
num_bph_packets 0 num_late_packets 0 num_early_packets 1
May  7 21:09:56.055 UTC: VOICE_PLAYOUT_DELAY_STATS(1/2): curr_playout_delay 3C
min_playout_delay 3C max_playout_delay 64 clock offset 197788E4
May  7 21:09:56.055 UTC: VOICE_PLAYOUT_ERROR(1/2): pred_conceal 0x0 inter_conceal 0x0
silence_conceal 0x0 buffer_overflow 0x1 endpt_det_error 0x0
May  7 21:09:56.855 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:57.907 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:09:57.907 UTC: FAX_RELAY_LINK_INFO_RSP_NTF: slot 1 port 2 timestamp 68137565
fr-entered (20ms)
May  7 21:09:57.907 UTC: chan_id [3/1:D] np_vsmgr_fax_relay_link_info_response:
May  7 21:10:15.047 UTC: np_fax_relay_t30_decode : Tx Direction
May  7 21:10:15.067 UTC: FARELAY_INIT_HS_MOD : 0xC
May  7 21:10:51.579 UTC: FAX_RELAY_DATA_PUMP_STATS(1/2) - valid:0x3FFC1F55 state_code:0x0
level:0x18 phase_jitter:0x5 freq_offset:0x0 eqm:0x7FFE jit_depth:0x230 jit_buf_ov:0x0
tx_paks:0x626 rx_pkts:0x5A inv_pkts:0x0 oos_pkts:0x0 hs_mod:0x8 init_hs_mod:0xC tx_pgs:0x1
 rx_pgs:0x0 ecm:0x1 nsf_country:0x0 nsf_manuf_len:0x20
nsf_manuf:0031B8EE80C48511DD0D0000DDDD0000DDDD00000000000000022ED00B0A400 encap:0x1
pkt_loss_con:0x0
May  7 21:10:52.463 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:10:52.463 UTC: vsm(1/2): np_vsmgr_voice_state_change - NULL DSP Interface Handle
```

debug nextport vsmgr detail Command on the Terminating Gateway

```
Router# debug nextport vsmgr detail
NextPort Voice Service Manager:
  NP Voice Service Manager Detail debugging is on
.
.
Router#
May  7 21:09:51.179 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:51.263 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:09:51.303 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:51.443 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:09:51.467 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:09:51.467 UTC: request_id = 0x01, request_type = 0x0F
May  7 21:09:51.467 UTC: VOICE_TRANSMIT_STATS(1/2): num_voice_packets 0 num_sig_packets 0
num_cn_packets 0 transmit_duration 0 end_point_detection 0
May  7 21:09:51.467 UTC: VOICE_RECEIVE_STATS(1/2): num_voice_packets 0 num_sig_packets 0
num_cn_packets 0 receive_duration 0  voice_receive_duration 0 num_pos_packets 0
num_bph_packets 0 num_late_packets 0 num_early_packets 0
May  7 21:09:51.467 UTC: VOICE_PLAYOUT_DELAY_STATS(1/2): curr_playout_delay 0
```

```
min_playout_delay 0 max_playout_delay 0 clock offset 0
May  7 21:09:51.467 UTC: VOICE_PLAYOUT_ERROR(1/2): pred_conceal 0x0 inter_conceal 0x0
silence_conceal 0x0 buffer_overflow 0x0 endpt_det_error 0x0
May  7 21:09:53.787 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:09:53.787 UTC: request_id = 0x01, request_type = 0x0F
May  7 21:09:53.787 UTC: VOICE_TRANSMIT_STATS(1/2): num_voice_packets 19 num_sig_packets 0
 num_cn_packets 1 transmit_duration 910 end_point_detection 0
May  7 21:09:53.787 UTC: VOICE_RECEIVE_STATS(1/2): num_voice_packets 1F num_sig_packets 0
num_cn_packets 2 receive_duration 910  voice_receive_duration 0 num_pos_packets 0
num_bph_packets 0 num_late_packets 0 num_early_packets 0
May  7 21:09:53.787 UTC: VOICE_PLAYOUT_DELAY_STATS(1/2): curr_playout_delay 5A
min_playout_delay 5A max_playout_delay 5A clock offset 68877C4
May  7 21:09:53.787 UTC: VOICE_PLAYOUT_ERROR(1/2): pred_conceal 0x0 inter_conceal 0x0
silence_conceal 0x0 buffer_overflow 0x0 endpt_det_error 0x0
May  7 21:09:56.571 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:09:56.571 UTC: request_id = 0x01, request_type = 0x0F
May  7 21:09:56.571 UTC: VOICE_TRANSMIT_STATS(1/2): num_voice_packets A5 num_sig_packets 0
 num_cn_packets 1 transmit_duration 13F6 end_point_detection 0
May  7 21:09:56.571 UTC: VOICE_RECEIVE_STATS(1/2): num_voice_packets 30 num_sig_packets 0
num_cn_packets 2 receive_duration 13F6  voice_receive_duration 7D0 num_pos_packets 0
num_bph_packets 0 num_late_packets 0 num_early_packets 0
May  7 21:09:56.571 UTC: VOICE_PLAYOUT_DELAY_STATS(1/2): curr_playout_delay 64
min_playout_delay 5A max_playout_delay 64 clock offset 68877D4
May  7 21:09:56.571 UTC: VOICE_PLAYOUT_ERROR(1/2): pred_conceal 0x0 inter_conceal 0x0
silence_conceal 0x0 buffer_overflow 0x0 endpt_det_error 0x0
May  7 21:09:56.807 UTC: VOICE_DET_STATUS_CHANGE_NTF(1/2): detector mask: 1 timestamp
791687D5
May  7 21:09:56.855 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:09:57.911 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:09:57.911 UTC: FAX_RELAY_LINK_INFO_RSP_NTF: slot 1 port 2 timestamp 65325022
fr-entered (20ms)
May  7 21:09:57.911 UTC: chan_id [3/1:D (6)] np_vsmgr_fax_relay_link_info_response:
May  7 21:10:15.043 UTC: np_fax_relay_t30_decode : Rx Direction
May  7 21:10:15.107 UTC: FARELAY_INIT_HS_MOD : 0x8
May  7 21:10:51.376 UTC: FAX_RELAY_DET_STATUS_CHANGE: slot: 1 port: 2 detector mask 0x2
May  7 21:10:51.404 UTC: FAX_RELAY_DATA_PUMP_STATS(1/2) - valid:0x3FFC1F55 state_code:0x1
level:0x18 phase_jitter:0x0 freq_offset:0x0 eqm:0x7FFE jit_depth:0x39E jit_buf_ov:0x0
tx_paks:0x5A rx_pkts:0x626 inv_pkts:0x0 oos_pkts:0x0 hs_mod:0x8 init_hs_mod:0x8 tx_pgs:0x0
 rx_pgs:0x1 ecm:0x1 nsf_country:0x0 nsf_manuf_len:0x20
nsf_manuf:0031B8EE80C48511DD0D0000DDDD0000DDDD00000000000000022ED00B0A400 encap:0x1
pkt_loss_con:0x0
May  7 21:10:52.288 UTC: FAX_RELAY_LINK_INFO_RSP_NTF: slot 1 port 2 timestamp 65760060
fr-end
May  7 21:10:52.304 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:10:52.388 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state ACTIVE
May  7 21:10:52.416 UTC: np_vsmgr_dispatch_voice_rsp(1/2): VOICE_LINK_INFO_RSP_NTF Received

May  7 21:10:52.416 UTC: request_id = 0x05, request_type = 0x30
May  7 21:10:52.416 UTC: VOICE_LEVELS_STATS(1/2): tx_power FF7E tx_mean FF7F rx_power FDBD
 rx_mean FB48  bnl FD81 erl FD acom 1EA tx_act 1 rx_act 0
May  7 21:10:52.440 UTC: vsm(1/2): np_vsmgr_voice_state_change() - state IDLE
May  7 21:10:52.440 UTC: vsm(1/2): np_vsmgr_voice_state_change - NULL DSP Interface Handle
```

Related Commands

Command	Description
debug dspapi detail	Displays details of the DSP API message events with debugging enabled.
voicecap entry	Creates a voicecap on NextPort platforms.
voicecap configure	Applies a voicecap on NextPort platforms.

debug nhrp

To enable Next Hop Resolution Protocol (NHRP) debugging, use the **debug nhrp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug nhrp [{**attribute** | **cache** | **condition**{**interface tunnel** *number* | **Virtual-Access** *number* | **peer**{**nbma**{*nbma-addressnbma-name*} | **tunnel**{*ipv4-address* | *ipv6-address*/*mask*}} | **vrf** *vrf-name*} | **detail** | **error** | **extension** | **group** | **packet** | **rate** | **routing**}]
no debug nhrp[{**attribute** | **cache** | **condition**{**interface tunnel** *number* | **Virtual-Access** *number* | **peer**{**nbma**{*nbma-addressnbma-name*} | **tunnel**{*ipv4-address* | *ipv6-address*/*mask*}} | **vrf** *vrf-name*} | **detail** | **error** | **extension** | **group** | **packet** | **rate** | **routing**}]

Syntax Description

attribute	(Optional) Enables NHRP attribute debugging operations.
cache	(Optional) Enables NHRP cache debugging operations.
condition	(Optional) Enables NHRP conditional debugging operations.
interface tunnel *number*	Enables debugging operations for the tunnel interface.
Virtual-Access *number*	Enables debugging operations for the virtual access interface.
nbma	Enables debugging operations for the non-broadcast multiple access (NBMA) network.
nbma-address	Enables debugging operations based on the IPv4 address of the NBMA network.
nbma-name	NBMA network name.
tunnel {*IPv4-address* | *IPv6-address*/*mask*}	Enables debugging operations for IPv4 or IPv6 addresses of the tunnel in the NBMA network.
vrf *vrf-name*	Enables debugging operations for the tunnel interface.
detail	(Optional) Displays detailed logs of NHRP debugs.
error	(Optional) Enables NHRP error debugging operations.
extension	(Optional) Enables NHRP extension processing debugging operations.
group	(Optional) Enables NHRP group debugging operations.
packet	(Optional) Enables NHRP activity debugging.
rate	(Optional) Enables NHRP rate limiting.
routing	(Optional) Enables NHRP routing debugging operations.

Command Default

NHRP debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(20)T	This command was introduced.
15.3(2)T	This command was modified. The **detail** keyword was added and the command output was enhanced to display more NHRP debugging information. The **Virtual-Access** *number* keyword-argument pair was added.

Usage Guidelines

Use the **debug nhrp detail** command to view the NHRP attribute logs.

The **Virtual-Access** *number* keyword-argument pair is visible only if the virtual access interface is available on the device.

Examples

The following example shows NHRP debugging output for IPv6:

```
Router# debug nhrp
Aug  9 13:13:41.486: NHRP: Attempting to send packet via DEST
   - 2001:0db8:3c4d:0015:0000:0000:1a2f:3d2c/32
Aug  9 13:13:41.486: NHRP: Encapsulation succeeded.
Aug  9 13:13:41.486: NHRP: Tunnel NBMA addr 11.11.11.99
Aug  9 13:13:41.486: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 105
Aug  9 13:13:41.486: src: 2001:0db8:3c4d:0015:0000:0000:1a2f:3d2c/32,
          dst: 2001:0db8:3c4d:0015:0000:0000:1a2f:3d2c/32
Aug  9 13:13:41.486: NHRP: 105 bytes out Tunnel0
Aug  9 13:13:41.486: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 125
```

The following example shows NHRP debugging output for IPv4:

```
Router# debug nhrp
Aug  9 13:13:41.486: NHRP: Attempting to send packet via DEST 10.1.1.99
Aug  9 13:13:41.486: NHRP: Encapsulation succeeded.  Tunnel IP addr 10.11.11.99
Aug  9 13:13:41.486: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 105
Aug  9 13:13:41.486:       src: 10.1.1.11, dst: 10.1.1.99
Aug  9 13:13:41.486: NHRP: 105 bytes out Tunnel0
Aug  9 13:13:41.486: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 125
Aug  9 13:13:41.486: NHRP: netid_in = 0, to_us = 1
```

The following example shows NHRP debugging output for the **detail** keyword:

```
Device# debug nhrp detail
NHRP detail debugging is on

*Aug  3 06:28:38.077: NHRP: if_up: Tunnel0 proto 'NHRP_IPv4'
*Aug  3 06:28:38.077: NHRP: Registration with Tunnels Decap Module succeeded
*Aug  3 06:28:38.077: NHRP: Adding all static maps to cache
*Aug  3 06:28:38.077: NHRP: Adding Tunnel Endpoints (VPN: 10.0.0.254, NBMA: 172.16.1.4)
*Aug  3 06:28:38.077: NHRP: Successfully attached NHRP subblock for Tunnel Endpoints (VPN:
 10.0.0.254, NBMA: 172.16.1.4)
*Aug  3 06:28:38.077: NHRP: if_up: Tunnel0 proto 'NHRP_IPv6'
*Aug  3 06:28:38.077: NHRP: Registration with Tunnels Decap Module succeeded
*Aug  3 06:28:38.077: NHRP: Adding all static maps to cache
*Aug  3 06:28:38.077: NHRP: Adding Tunnel Endpoints (VPN: 2001::2, NBMA: 172.16.1.4)
*Aug  3 06:28:38.078: NHRP: Successfully attached NHRP subblock for Tunnel Endpoints (VPN:
 2001::2, NBMA: 172.16.1.4)

*Aug  3 06:28:38.078: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Aug  3 06:28:38.079: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

```
*Aug  3 06:28:38.716: %SYS-5-CONFIG_I: Configured from console by console
*Aug  3 06:28:39.030: NHRP: Sending one-time request for nhs 2001::2
*Aug  3 06:28:39.030: NHRP-ATTR:  Requester Ext Len: Total ext_len  with NHRP attribute VPE
 64

*Aug  3 06:28:39.030: NHRP: Attempting to send packet through interface Tunnel0 via dst
2001::2
*Aug  3 06:28:39.030: NHRP: Encapsulation succeeded.  Sending NHRP Control Packet  NBMA
Address: 172.16.1.4
*Aug  3 06:28:39.030: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 160
*Aug  3 06:28:39.030:       src: 2001::3, dst: 2001::2
*Aug  3 06:28:39.031: NHRP: 188 bytes out Tunnel0
*Aug  3 06:28:39.032: NHRP-ATTR: ext_type: 32771, ext_len : 32

*Aug  3 06:28:39.032: NHRP-ATTR: ext_type: 32772, ext_len : 0

*Aug  3 06:28:39.032: NHRP-ATTR: ext_type: 32773, ext_len : 0

*Aug  3 06:28:39.032: NHRP-ATTR: ext_type: 32775, ext_len : 8

*Aug  3 06:28:39.032: NHRP-ATTR: ext_type: 9, ext_len : 32

*Aug  3 06:28:39.032: NHRP-ATTR: ext_type: 32768, ext_len : 0

*Aug  3 06:28:39.032: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 224
*Aug  3 06:28:39.032: NHRP: netid_in = 0, to_us = 1
*Aug  3 06:28:39.032: NHRP: NHS-UP: 2001::2
*Aug  3 06:28:39.032: NHRP: Adding Tunnel Endpoints (VPN: FE80::A8BB:CCFF:FE01:F500, NBMA:
 172.16.1.4)
*Aug  3 06:28:39.032: NHRP: Successfully attached NHRP subblock for Tunnel Endpoints (VPN:
 FE80::A8BB:CCFF:FE01:F500, NBMA: 172.16.1.4)
*Aug  3 06:28:39.032: NHRP: Caching Additional Address: FE80::A8BB:CCFF:FE01:F500, cache:
0x0x2A98CBCE28, hold_time: 300
*Aug  3 06:28:39.060: NHRP: Sending one-time request for nhs 10.0.0.254
*Aug  3 06:28:39.060: NHRP-ATTR:  Requester Ext Len: Total ext_len  with NHRP attribute VPE
 52

*Aug  3 06:28:39.060: NHRP: Attempting to send packet through interface Tunnel0 via dst
10.0.0.254
*Aug  3 06:28:39.060: NHRP: Encapsulation succeeded.  Sending NHRP Control Packet  NBMA
Address: 172.16.1.4
*Aug  3 06:28:39.060: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 104
*Aug  3 06:28:39.060:       src: 10.0.0.2, dst: 10.0.0.254
*Aug  3 06:28:39.060: NHRP: 132 bytes out Tunnel0
*Aug  3 06:28:39.061: NHRP-ATTR: ext_type: 32771, ext_len : 20

*Aug  3 06:28:39.061: NHRP-ATTR: ext_type: 32772, ext_len : 0

*Aug  3 06:28:39.061: NHRP-ATTR: ext_type: 32773, ext_len : 0

*Aug  3 06:28:39.061: NHRP-ATTR: ext_type: 32775, ext_len : 8

*Aug  3 06:28:39.061: NHRP-ATTR: ext_type: 9, ext_len : 20

*Aug  3 06:28:39.061: NHRP-ATTR: ext_type: 32768, ext_len : 0

*Aug  3 06:28:39.061: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 124
*Aug  3 06:28:39.061: NHRP: netid_in = 0, to_us = 1
*Aug  3 06:28:39.061: NHRP: NHS-UP: 10.0.0.254
*Aug  3 06:28:39.080: NHRP-ATTR:  Requester Ext Len: Total ext_len  with NHRP attribute VPE
 52

*Aug  3 06:28:39.080: NHRP: Attempting to send packet through interface Tunnel0 via dst
10.0.0.254
```

```
*Aug  3 06:28:39.080: NHRP: Encapsulation succeeded.  Sending NHRP Control Packet  NBMA
Address: 172.16.1.4
*Aug  3 06:28:39.080: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 104
*Aug  3 06:28:39.080:       src: 10.0.0.2, dst: 10.0.0.254
*Aug  3 06:28:39.080: NHRP: 132 bytes out Tunnel0
*Aug  3 06:28:39.080: NHRP-ATTR:  Requester Ext Len: Total ext_len  with NHRP attribute VPE
 64

*Aug  3 06:28:39.080: NHRP: Attempting to send packet through interface Tunnel0 via dst
2001::2
*Aug  3 06:28:39.081: NHRP: Encapsulation succeeded.  Sending NHRP Control Packet  NBMA
Address: 172.16.1.4
*Aug  3 06:28:39.081: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 160
*Aug  3 06:28:39.081:       src: 2001::3, dst: 2001::2
*Aug  3 06:28:39.081: NHRP: 188 bytes out Tunnel0
*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32771, ext_len : 20

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32772, ext_len : 0

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32773, ext_len : 0

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32775, ext_len : 8

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 9, ext_len : 20

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32768, ext_len : 0

*Aug  3 06:28:39.081: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 124
*Aug  3 06:28:39.081: NHRP: netid_in = 0, to_us = 1
*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32771, ext_len : 32

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32772, ext_len : 0

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32773, ext_len : 0

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32775, ext_len : 8

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 9, ext_len : 32

*Aug  3 06:28:39.081: NHRP-ATTR: ext_type: 32768, ext_len : 0

*Aug  3 06:28:39.081: NHRP: Receive Registration Reply via Tunnel0

vrf 0, packet size: 224
*Aug  3 06:28:39.082: NHRP: netid_in = 0, to_us = 1
*Aug  3 06:28:39.082: NHRP: Adding Tunnel Endpoints (VPN: FE80::A8BB:CCFF:FE01:F500, NBMA:
 172.16.1.4)
*Aug  3 06:28:39.082: NHRP: NHRP subblock already exists for Tunnel Endpoints (VPN:
FE80::A8BB:CCFF:FE01:F500, NBMA: 172.16.1.4)
*Aug  3 06:28:39.082: NHRP: Cache already has a subblock node attached for Tunnel Endpoints
 (VPN: FE80::A8BB:CCFF:FE01:F500, NBMA: 172.16.1.4)
*Aug  3 06:28:39.082: NHRP: Caching Additional Address: FE80::A8BB:CCFF:FE01:F500, cache:
0x0x2A98CBCE28, hold_time: 300
*Aug  3 06:28:40.080: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state
 to up
*Aug  3 06:28:40.081: NHRP: if_up: Tunnel0 proto 'NHRP_IPv4'
*Aug  3 06:28:40.081: NHRP: Registration with Tunnels Decap Module succeeded
*Aug  3 06:28:40.081: NHRP: Adding all static maps to cache
*Aug  3 06:28:40.081: NHRP: Adding Tunnel Endpoints (VPN: 10.0.0.254, NBMA: 172.16.1.4)
*Aug  3 06:28:40.081: NHRP: NHRP subblock already exists for Tunnel Endpoints (VPN:
10.0.0.254, NBMA: 172.16.1.4)
*Aug  3 06:28:40.081: NHRP: Cache already has a subblock node attached for Tunnel Endpoints
 (VPN: 10.0.0.254, NBMA: 172.16.1.4)
*Aug  3 06:28:40.081: NHRP-ATTR:  Requester Ext Len: Total ext_len  with NHRP attribute VPE
```

```
 52

*Aug  3 06:28:40.081: NHRP: Attempting to send packet through interface Tunnel0 via dst
10.0.0.254
*Aug  3 06:28:40.081: NHRP: Encapsulation succeeded.  Sending NHRP Control Packet  NBMA
Address: 172.16.1.4
*Aug  3 06:28:40.081: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 104
*Aug  3 06:28:40.081:       src: 10.0.0.2, dst: 10.0.0.254
*Aug  3 06:28:40.081: NHRP: 132 bytes out Tunnel0
*Aug  3 06:28:40.081: NHRP: if_up: Tunnel0 proto 'NHRP_IPv6'
*Aug  3 06:28:40.081: NHRP: Registration with Tunnels Decap Module succeeded
*Aug  3 06:28:40.081: NHRP: Adding all static maps to cache
*Aug  3 06:28:40.081: NHRP: Adding Tunnel Endpoints (VPN: 2001::2, NBMA: 172.16.1.4)
*Aug  3 06:28:40.081: NHRP: NHRP subblock already exists for Tunnel Endpoints (VPN: 2001::2,
 NBMA: 172.16.1.4)
*Aug  3 06:28:40.081: NHRP: Cache already has a subblock node attached for Tunnel Endpoints
 (VPN: 2001::2, NBMA: 172.16.1.4)
*Aug  3 06:28:40.081: NHRP-ATTR:  Requester Ext Len: Total ext_len  with NHRP attribute VPE
 64

*Aug  3 06:28:40.081: NHRP: Attempting to send packet through interface Tunnel0 via dst
2001::2
*Aug  3 06:28:40.081: NHRP: Encapsulation succeeded.  Sending NHRP Control Packet  NBMA
Address: 172.16.1.4
*Aug  3 06:28:40.081: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 160
*Aug  3 06:28:40.081:       src: 2001::3, dst: 2001::2
*Aug  3 06:28:40.081: NHRP: 188 bytes out Tunnel0
*Aug  3 06:28:40.081: %LINK-3-UPDOWN: Interface Tunnel0, changed state to up
*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32771, ext_len : 20

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32772, ext_len : 0

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32773, ext_len : 0

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32775, ext_len : 8

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 9, ext_len : 20

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32768, ext_len : 0

*Aug  3 06:28:40.084: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 124
*Aug  3 06:28:40.084: NHRP: netid_in = 0, to_us = 1
*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32771, ext_len : 32

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32772, ext_len : 0

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32773, ext_len : 0

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32775, ext_len : 8

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 9, ext_len : 32

*Aug  3 06:28:40.084: NHRP-ATTR: ext_type: 32768, ext_len : 0

*Aug  3 06:28:40.084: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 224
*Aug  3 06:28:40.084: NHRP: netid_in = 0, to_us = 1
*Aug  3 06:28:40.084: NHRP: Adding Tunnel Endpoints (VPN: FE80::A8BB:CCFF:FE01:F500, NBMA:
 172.16.1.4)
*Aug  3 06:28:40.084: NHRP: NHRP subblock already exists for Tunnel Endpoints (VPN:
FE80::A8BB:CCFF:FE01:F500, NBMA: 172.16.1.4)
*Aug  3 06:28:40.084: NHRP:
Cache already has a subblock node attached for Tunnel Endpoints (VPN:
FE80::A8BB:CCFF:FE01:F500, NBMA: 172.16.1.4)
*Aug  3 06:28:40.084: NHRP: Caching Additional Address: FE80::A8BB:CCFF:FE01:F500, cache:
```

```
0x0x2A98CBCE28, hold_time: 300

*Aug  3 06:28:41.521: %DUAL-5-NBRCHANGE: EIGRP-IPv6 2: Neighbor FE80::A8BB:CCFF:FE01:F500
(Tunnel0) is up: new adjacency

*Aug  3 06:28:41.531: NHRP: Attempting to check and send Traffic Indication to NBMA: UNKNOWN
*Aug  3 06:28:41.531: NHRP: IPv6 NHRP Shortcut Enabled: Attempting switching
*Aug  3 06:28:41.570: NHRP: Attempting to check and send Traffic Indication to NBMA: UNKNOWN
*Aug  3 06:28:41.570: NHRP: IPv6 NHRP Shortcut Enabled: Attempting switching
*Aug  3 06:28:41.590: NHRP: Attempting to check and send Traffic Indication to NBMA: UNKNOWN
*Aug  3 06:28:41.590: NHRP: IPv6 NHRP Shortcut Enabled: Attempting switching
*Aug  3 06:28:41.610: NHRP: Attempting to check and send Traffic Indication to NBMA: UNKNOWN
*Aug  3 06:28:41.610: NHRP: IPv6 NHRP Shortcut Enabled: Attempting switching
*Aug  3 06:28:42.731: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.254 (Tunnel0) is up:
 new adjacency

*Aug  3 06:28:43.140: NHRP: Attempting to check and send Traffic Indication to NBMA: UNKNOWN
*Aug  3 06:28:43.140: NHRP: IPv6 NHRP Shortcut Enabled: Attempting switching
```

Related Commands

Command	Description
debug dmvpn	Displays DMVPN session debugging information.
debug nhrp error	Displays NHRP error-level debugging information.

debug nhrp condition

To enable Next Hop Resolution Protocol (NHRP) conditional debugging, use the **debug nhrp condition**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug nhrp condition [{**interface tunnel** *number* | **peer** {**nbma** {*ip-addressFQDN-string*} | **tunnel** {*ip-addressipv6-address*}} | **vrf** *vrf-name*}]
no debug nhrp condition [{**interface tunnel** *number* | **peer** {**nbma** {*ip-addressFQDN-string*} | **tunnel** {*ip-addressipv6-address*}} | **vrf** *vrf-name*}]

Syntax Description

tunnel	(Optional) Specifies a tunnel.
interface	(Optional) Displays NHRP information based on a specific interface.
tunnel *number*	(Optional) Specifies the tunnel address for the NHRP peer.
peer	(Optional) Specifies an NHRP peer.
nbma	(Optional) Specifies mapping nonbroadcast multiple access (NBMA).
ip-address	(Optional) The IPv4 address for the NHRP peer.
FQDN-string	(Optional) Next hop server (NHS) fully qualified domain name (FQDN) string.
ipv6-address	(Optional) The IPv6 address for the NHRP peer. **Note** Cisco IOS XE Release 2.5 does not support the ipv6-address argument.
vrf *vrf-name*	(Optional) Specifies debugging information for sessions related to the specified virtual routing and forwarding (VRF) configuration.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.
12.4(20)T	This command was modified. The *ipv6-address* argument was added.
Cisco IOS XE Release 2.5	This command was integrated into Cisco IOS XE Release 2.5 and implemented on the Cisco ASR 1000 Series Aggregation Services Routers.
15.1(2)T	This command was modified. The *FQDN-string* argument was added.

Examples

The following example shows how to enable conditional NHRP debugging for a specified NBMA address:

```
Router# debug nhrp condition peer tunnel 192.0.2.1
```

The following example shows how to enable conditional NHRP debugging for a specified FQDN string:

```
Router# debug nhrp condition peer examplehub.example1.com
```

Related Commands

Command	Description
debug dmvpn	Displays DMVPN session debugging information.
debug nhrp error	Displays NHRP error level debugging information.

debug nhrp error

To display Next Hop Resolution Protocol (NHRP) error-level debugging information, use the **debug nhrp error**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug nhrp {**ipv4** | **ipv6**} **error**
no debug nhrp {**ipv4** | **ipv6**} **error**

Syntax Description

ipv4	Specifies the IPv6 overlay network.
ipv6	Specifies the IPv6 overlay network. **Note** Cisco IOS XE Release 2.5 does not support the ipv6 keyword.

Command Default

NHRP error-level debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.
12.4(20)T	The **ipv4** and **ipv6** keywords were added.
Cisco IOS XE Release 2.5	This command was modified. It was integrated into Cisco IOS XE Release 2.5.

Examples

The following example shows how to enable error level debugging for IPv4 NHRP:

```
Router# debug nhrp ipv4 error
NHRP errors debugging is on
```

Related Commands

Command	Description
debug dmvpn	Displays DMVPN session debugging information.
debug nhrp condition	Enables NHRP conditional debugging.

debug nhrp extension

To display the extensions portion of a NHRP packet, use the **debug nhrp extension**privileged EXEC command. The **no** form of this command disables debugging output.

NHRP : debug nhrp extension commanddebug nhrp extension commanddebug nhrp extension no debug nhrp extension

Syntax Description

This command has no arguments or keywords.

Command History

Release	Modification
10.0	This command was introduced.
12.2(13)T	This command is no longer supported in Cisco IOS Mainline or Technology-based (T) releases. It may continue to appear in Cisco IOS 12.2S-family releases.

Examples

The following is sample output from the **debug nhrp extension**command:

```
Router# debug nhrp extension
NHRP extension processing debugging is on
Router#
Forward Transit NHS Record Extension(4):
 (C-1) code: no error(0)
       prefix: 0, mtu: 9180, hd_time: 7200
       addr_len: 20(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 0
       client NBMA: 47.0091810000000002ba08e101.525354555354.01
       client protocol: 135.206.58.54
Reverse Transit NHS Record Extension(5):
Responder Address Extension(3):
 (C) code: no error(0)
       prefix: 0, mtu: 9180, hd_time: 7200
       addr_len: 20(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 0
       client NBMA: 47.0091810000000002ba08e101.525354555355.01
       client protocol: 135.206.58.55
Forward Transit NHS Record Extension(4):
 (C-1) code: no error(0)
       prefix: 0, mtu: 9180, hd_time: 7200
       addr_len: 20(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 0
       client NBMA: 47.0091810000000002ba08e101.525354555354.01
       client protocol: 135.206.58.54
Reverse Transit NHS Record Extension(5):
Responder Address Extension(3):
Forward Transit NHS Record Extension(4):
Reverse Transit NHS Record Extension(5):
```

debug nhrp options

To display information about NHRP option processing, use the **debugnhrpoptions**privileged EXEC command. The **no** form of this command disables debugging output.

debug nhrp options
no debug nhrp options

Syntax Description

This command has no arguments or keywords.

Command History

Release	Modification
10.0	This command was introduced.
12.2(13)T	This command is no longer supported in Cisco IOS Mainline or Technology-based (T) releases. It may continue to appear in Cisco IOS 12.2S-family releases.

Usage Guidelines

Use this command to show you whether there are problems or error situations with NHRP option processing (for example, unknown options).

Examples

The following is sample output from the **debugnhrpoptions**command:

```
Router#
debug nhrp options
NHRP-OPT: MASK 4
NHRP-OPT-MASK: FFFFFFFF
NHRP-OPT: NETID 4
NHRP-OPT: RESPONDER 4
NHRP-OPT: RECORD 0
NHRP-OPT: RRECORD 0
```

The following table describes the significant fields shown in the display.

Table 39: debug nhrp options Field Descriptions

Field	Descriptions
NHRP-OPT	NHRP options debugging output.
MASK 4	Number of bytes of information in the destination prefix option.
NHRP-OPT-MASK	Contents of the destination prefix option.
NETID	Number of bytes of information in the subnetwork identifier option.
RESPONDER	Number of bytes of information in the responder address option.
RECORD	Forward record option.
RRECORD	Reverse record option.

Related Commands

Command	Description
debug nhrp	Displays information about NHRP activity.
debug nhrp packet	Displays a dump of NHRP packets.

debug nhrp packet

To display a dump of NHRP packets, use the **debug nhrp packet**privileged EXEC command. The **no** form of this command disables debugging output.

NHRP : debug nhrp packet commanddebug nhrp packet commanddebug nhrp packet
no NHRP : debug nhrp packet commanddebug nhrp packet commanddebug nhrp packet

Syntax Description

This command has no arguments or keywords.

Command History

Release	Modification
10.0	This command was introduced.
12.2(13)T	This command is no longer supported in Cisco IOS Mainline or Technology-based (T) releases. It may continue to appear in Cisco IOS 12.2S-family releases.

Examples

The following is sample output from the **debug nhrp packet**command:

```
Router# debug nhrp packet
NHRP activity debugging is on
Router#
NHRP: Send Purge Request via ATM3/0.1, packet size: 72
 src: 135.206.58.55, dst: 135.206.58.56
 (F) afn: NSAP(3), type: IP(800), hop: 255, ver: 1
     shtl: 20(NSAP), sstl: 0(NSAP)
 (M) flags: "reply required", reqid: 2
     src NBMA: 47.0091810000000002ba08e101.525354555355.01
     src protocol: 135.206.58.55, dst protocol: 135.206.58.56
 (C-1) code: no error(0)
       prefix: 0, mtu: 9180, hd_time: 0
       addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 0
       client protocol: 135.206.58.130
NHRP: Receive Purge Reply via ATM3/0.1, packet size: 72
 (F) afn: NSAP(3), type: IP(800), hop: 254, ver: 1
     shtl: 20(NSAP), sstl: 0(NSAP)
 (M) flags: "reply required", reqid: 2
     src NBMA: 47.0091810000000002ba08e101.525354555355.01
     src protocol: 135.206.58.55, dst protocol: 135.206.58.56
 (C-1) code: no error(0)
       prefix: 0, mtu: 9180, hd_time: 0
       addr_len: 0(NSAP), subaddr_len: 0(NSAP), proto_len: 4, pref: 0
       client protocol: 135.206.58.130
```

debug nhrp rate

To display information about NHRP traffic rate limits, use the **debugnhrprate**privileged EXEC command. The **no** form of this command disables debugging output.

debug nhrp rate
no debug nhrp rate

Syntax Description

This command has no arguments or keywords.

Command History

Release	Modification
10.0	This command was introduced.
12.2(13)T	This command is no longer supported in Cisco IOS Mainline or Technology-based (T) releases. It may continue to appear in Cisco IOS 12.2S-family releases.

Usage Guidelines

Use this command to verify that the traffic is consistent with the setting of the NHRP commands (such as **ipnhrpuse** and**ipmax-send** commands).

Examples

The following is sample output from the **debugnhrprate**command:

```
Router#
debug nhrp rate
NHRP-RATE: Sending initial request
NHRP-RATE: Retransmitting request (retrans ivl 2)
NHRP-RATE: Retransmitting request (retrans ivl 4)
NHRP-RATE: Ethernet1: Used 3
```

The following table describes the significant fields shown in the display.

Table 40: debug nhrp rate Field Descriptions

Field	Descriptions
NHRP-RATE	NHRP rate debugging output.
Sending initial request	First time an attempt was made to send an NHRP packet to a particular destination.
Retransmitting request	Indicates that the NHRP packet was re-sent, and shows the time interval (in seconds) to wait before the NHRP packet is re-sent again.
Ethernet1: Used 3	Interface over which the NHRP packet was sent. Number of packets sent out of the default maximum five (in this case, three were sent).

Related Commands

Command	Description
debug nhrp	Displays information about NHRP activity.

Command	Description
debug nhrp options	Displays information about NHRP option processing

debug ntp

To display debugging messages for Network Time Protocol (NTP) features, use the **debug ntp**command in prvileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ntp {**adjust** | **all** | **authentication** | **core** | **events** | **loopfilter** | **packet** | **params** | **refclock** | **select** | **sync** | **validity**}
no debug ntp {**adjust** | **all** | **authentication** | **core** | **events** | **loopfilter** | **packet** | **params** | **refclock** | **select** | **sync** | **validity**}

Syntax Description

adjust	Displays debugging information on NTP clock adjustments.
all	Displays all debugging information on NTP.
authentication	Displays debugging information on NTP authentication.
core	Displays debugging information on NTP core messages.
events	Displays debugging information on NTP events.
loopfilter	Displays debugging information on NTP loop filters.
packet	Displays debugging information on NTP packets.
params	Displays debugging information on NTP clock parameters.
refclock	Displays debugging information on NTP reference clocks.
select	Displays debugging information on NTP clock selection.
sync	Displays debugging information on NTP clock synchronization.
validity	Displays debugging information on NTP peer clock validity.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.1	This command was introduced in a release prior to Cisco IOS Release 12.1.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	Support for IPv6 and NTP version 4 was added. The **all** and **core** keywords were added. The **authentication**, **loopfilter**, **params**, **select**, **sync** and **validity** keywords were removed. The **packets** keyword was modified as **packet**.
Cisco IOS XE Release 3.5S	This command was integrated into Cisco IOS XE Release 3.5S.
Cisco IOS Release 15.2(1)S	This command was integrated into Cisco IOS Release 15.2(1)S.

Usage Guidelines

Starting from Cisco IOS Release 12.4(20)T, NTP version 4 is supported. In NTP version 4 the debugging options available are **adjust**, **all**, **core**, **events**, **packet**, and **refclock**. In NTP version 3 the debugging options available were **events**, **authentication**, **loopfilter**, **packets**, **params**, **select**, **sync** and **validity**.

Examples

The following example shows how to enable all debugging options for NTP:

```
Router# debug ntp all
NTP events debugging is on
NTP core messages debugging is on
NTP clock adjustments debugging is on
NTP reference clocks debugging is on
NTP packets debugging is on
```

Related Commands

Command	Description
ntp refclock	Configures an external clock source for use with NTP services.

debug oam

To display operation and maintenance (OAM) events, use the **debugoam** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug oam
no debug oam

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following is sample output from the **debugoam** command:

```
Router# debug oam
4/0(O): VCD:0x0 DM:0x300 *OAM Cell* Length:0x39
0000 0300 0070 007A 0018 0100 0000 05FF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FF6A 6A6A 6A6A 6A6A 6A6A 6A6A 6A6A 6A6A 6A00 0000
```

The following table describes the significant fields shown in the display.

Table 41: debug oam Field Descriptions

Field	Description
0000	Virtual circuit designator (VCD) Special OAM indicator.
0300	Descriptor MODE bits for the ATM Interface Processor (AIP).
0	GFC (4 bits).
07	Virtual path identifier (VPI) (8 bits).
0007	Virtual channel identifier (VCI)(16 bits).
A	Payload type field (PTI) (4 bits).
00	Header Error Correction (8 bits).
1	OAM Fault mangement cell (4 bits).
8	OAM LOOPBACK indicator (4 bits).
01	Loopback indicator value, always 1 (8 bits).
00000005	Loopback unique ID, sequence number (32 bits).
FF6A	Fs and 6A required in the remaining cell, per UNI3.0.

debug object-group event

To enable debug messages for object-group events, use the **debug object-group event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug object-group event

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(1)S	This command was introduced in Cisco IOS Release 15.2(1)S.
Cisco IOS XE Release 3.5	This command was introduced in Cisco IOS XE Release 3.5.

Usage Guidelines

When an object group is created to identify traffic coming from a specific user or endpoint, object-group identity mode is entered where a security group can be specified for the object group with a security group tag (SGT) ID. The SGT ID is used by a Security Group Access (SGA) Zone-Based Policy firewall (ZBPF) to apply an enforcement policy by filtering on this SGT ID. The **debug object-group event** command is used to view messages for object-group events while configuring the class map part of the SGA ZBPF.

Note A policy map must also be configured for the SGA ZBPF.

Examples

The following is sample output from the **debug object-group event** command:

```
Router# debug object-group event
Router# configure terminal
Router(config)# object-group security objsgt1
Router(config-security-group)# GLO INFO conf_objectgroup_cmd type(3) name(objsgt1)
Router(config-security-group)# security-group tag 120
Router(config-security-group)#
*Nov 21 16:23:02.041: INFO og_security_create_fn
*Nov 21 16:23:02.041: og_security_sgt_copy_fn:1633: object_group 'objsgt1' sgt name '' id
120
*Nov 21 16:23:02.041: og_classes_update:1373: walking class-maps in object_group 'objsgt1'
Router(config-security-group)#exit
Router(config)#
Router(config)# object-group security objsgt2
Router(config-security-group)# GLO INFO conf_objectgroup_cmd type(3) name(objsgt2)gr
Router(config-security-group)# group-object objsgt1
Router(config-security-group)#
*Nov 21 16:23:44.891: INFO og_security_create_fn
*Nov 21 16:23:44.891: og_classes_update:1373: walking class-maps in object_group 'objsgt2'
Router(config-security-group)#exit
```

Related Commands

Command	Description
group-object	Specifies a nested reference to a type of user group.
match group-object security	Matches traffic from a user in the security group.
object-group security	Creates an object group to identify traffic coming from a specific user or endpoint.
security-group	Specifies the membership of security group for an object group.
show object-group	Displays the content of all user groups.

debug oer api

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer api** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display Optimized Edge Routing (OER) application interface debugging information, use the **debug oer api**command in privileged EXEC mode. To stop the display of OER application interface debugging information, use the **no** form of this command.

debug oer api [**detail**]
no debug oer api

Syntax Description

detail	(Optional) Displays detailed application interface debugging information.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(15)T	This command was introduced.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer api**command is used to display messages about any configured OER application interface providers or host devices. The OER application interface defines the mode of communication and messaging between applications and the network for the purpose of optimizing the traffic associated with the applications. A provider is defined as an entity outside the network in which the router configured as an OER master controller exists, for example, an ISP, or a branch office of the same company. The provider has one or more host devices running one or more applications that use the OER application interface to communicate with an OER master controller. A provider must be registered with an OER master controller before an application on a host device can interface with OER. Use the **api provider** command to register the provider, and use the **host-address**command to configure a host device. After registration, a host device in the provider network can initiate a session with an OER master controller. The application interface provides an automated method for networks to be aware of applications and provides application-aware performance routing.

Caution When the **detail** keyword is entered, the amount of detailed output to be displayed can utilize a considerable amount of system resources. Use the **detail**keyword with caution in a production network.

Examples

The following example enables the display of OER application interface debugging messages and the output shows that an OER policy failed due to a prefix that is not found:

```
Router# debug oer api
```

```
OER api debugging is on
*May 26 01:04:07.278: OER API: Data set id received 5, data set len 9, host ip 10.3.3.3,
session id 1, requies2
*May 26 01:04:07.278: OER API: Received get current policy, session id 1 request id 22
*May 26 01:04:07.278: OER API: Recvd Appl with Prot 256 DSCP 0 SrcPrefix 0.0.0.0/0
SrcMask 0.0.0.0
*May 26 01:04:07.278: OER API:  DstPrefix 10.2.0.0/24 DstMask 255.255.255.0 Sport_min 0
Sport_max 0 Dport_mi0
*May 26 01:04:07.278: OER API: get prefix policy failed - prefix not found
*May 26 01:04:07.278: OER API: Get curr policy cmd received. rc 0
*May 26 01:04:07.278: OER API: Received send status response, status 0, session id 1,
request id 22, sequence0
*May 26 01:04:07.278: OER API: rc for data set 0
```

The table below describes the significant fields shown in the display. The content of the debugging messages depends on the commands that are subsequently entered at the router prompt.

Table 42: debug oer api Field Descriptions

Field	Description
OER api debugging is on	Shows that application interface debugging is enabled.
OER API	Displays an OER application interface message.

Related Commands

Command	Description
api provider	Registers an application interface provider with an OER master controller and enters OER master controller application interface provider configuration mode.
host-address	Configures information about a host device used by an application interface provider to communicate with an OER master controller.
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.
show oer api provider	Displays information about application interface providers registered with OER.

debug oer api client

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer api** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

Note Effective with Cisco IOS Release 12.4(15)T, the **debug oer api client** command is replaced by the **debug oer api** command. See the **debug oer api** command for more information.

To display Optimized Edge Routing (OER) application interface client debugging information for master controller and border router communication, use the **debug oer api client** command in privileged EXEC mode. To stop the display of OER application interface debugging information, use the **no** form of this command.

debug oer api client [**detail**]
no debug oer api client [**detail**]

Syntax Description

detail	(Optional) Displays detailed information.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.4(15)T	The **debug oer api client** command is replaced by the **debug oer api** command.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer api client** command can be entered on a master controller. This command is used to display messages about a configured OER application interface client. When the **detail** keyword is entered, the amount of detailed output to be displayed can utilize a considerable amount of system resources. Use the **detail**keyword with caution in a production network.

Cisco IOS Release 12.4(15)T

In Cisco IOS Release 12.4(15)T and later releases, the **debug oer api client** command is replaced by the **debug oer api** command. The **debug oer api client** command is currently supported for backwards compatibility, but support may be removed in a future Cisco IOS software release.

Examples

The following example enables the display of OER application interface client debugging messages:

```
Router# debug oer api client
API Client debugging enabled
```

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer border

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer border** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display general OER border router debugging information, use the **debug oer border**command in privileged EXEC mode. To stop the display of OER debugging information, use the **no** form of this command.

debug oer border
no debug oer border

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer border** command is entered on a border router. This command is used to display debugging information about the OER border process, controlled routes and monitored prefixes.

Examples

The following example displays general OER debugging information:

```
Router# debug oer border
*May  4 22:32:33.695: OER BR: Process Message, msg 4, ptr 33272128, value  140
*May  4 22:32:34.455: OER BR: Timer event, 0
```

The table below describes the significant fields shown in the display.

Table 43: debug oer border Field Descriptions

Field	Description
OER BR:	Indicates debugging information for OER Border process.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer border active-probe

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer border active-probe** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display debugging information for active probes configured on the local border router, use the **debug oer border active-probe**command in privileged EXEC mode. To stop the display of debug event information, use the **no** form of this command.

debug oer border active-probe
no debug oer border active-probe

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer border active-probe** command is entered on a master controller. This command is used to display the status and results of active probes that are configured on the local border router.

Examples

The following example enables the display of active-probe debug information on a border router:

```
Router# debug oer border active-probe

*May  4 23:47:45.633: OER BR ACTIVE PROBE: Attempting to retrieve Probe
Statistics.
      probeType = echo, probeTarget = 10.1.5.1, probeTargetPort = 0
      probeSource = Default, probeSourcePort = 0, probeNextHop = Default
      probeIfIndex = 13
*May  4 23:47:45.633: OER BR ACTIVE PROBE: Completed retrieving Probe
Statistics.
      probeType = echo, probeTarget = 10.1.5.1, probeTargetPort = 0
      probeSource = Default, probeSourcePort = 0, probeNextHop = 10.30.30.2
      probeIfIndex = 13, SAA index = 15
*May  4 23:47:45.633: OER BR ACTIVE PROBE: Completions 11, Sum of rtt 172,
Max rtt 36, Min rtt 12
*May  4 23:47:45.693: OER BR ACTIVE PROBE: Attempting to retrieve Probe
Statistics.
```

```
      probeType = echo, probeTarget = 10.1.4.1, probeTargetPort = 0
      probeSource = Default, probeSourcePort = 0, probeNextHop = Default
      probeIfIndex = 13
*May  4 23:47:45.693: OER BR ACTIVE PROBE: Completed retrieving Probe
Statistics.
      probeType = echo, probeTarget = 10.1.4.1, probeTargetPort = 0
      probeSource = Default, probeSourcePort = 0, probeNextHop = 10.30.30.2
      probeIfIndex = 13, SAA index = 14
```

The table below describes the significant fields shown in the display.

Table 44: debug oer border active-probe Field Descriptions

Field	Description
OER BR ACTIVE PROBE:	Indicates debugging information for OER active probes on a border router.
Statistics	The heading for OER active probe statistics.
probeType	The active probe type. The active probe types that can be displayed are ICMP, TCP, and UDP.
probeTarget	The target IP address of the active probe.
probeTargetPort	The target port of the active probe.
probeSource	The source IP address of the active probe. Default is displayed for a locally generated active probe.
probeSourcePort	The source port of the active probe.
probeNextHop	The next hop for the active probe.
probeIfIndex	The active probe source interface index.
SAA index	The IP SLAs collection index number.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer border learn

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer border learn** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display debugging information about learned prefixes on the local border router, use the **debug oer border learn**command in privileged EXEC mode. To stop the display of debug event information, use the **no** form of this command.

debug oer border learn [**top** *number*]
no debug oer border learn [**top** *number*]

Syntax Description

top *number*	(Optional) Displays debugging information about the top delay or top throughput prefixes. The number of top delay or throughput prefixes can be specified. The range of prefixes that can be specified is a number from 1 to 65535.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer border learn** command is entered on a border router. This command is used to display debugging information about prefixes learned on the local border router.

Examples

The following example enables the display of active-probe debug information on a border router:

```
Router# debug oer border learn

*May  4 22:51:31.971: OER BR LEARN: Reporting prefix 1: 10.1.5.0, throughput 201
*May  4 22:51:31.971: OER BR LEARN: Reporting 1 throughput learned prefixes
*May  4 22:51:31.971: OER BR LEARN: State change, new STOPPED, old STARTED, reason Stop
Learn
```

The table below describes the significant fields shown in the display.

Table 45: debug oer border learn Field Descriptions

Field	Description
OER BR LEARN:	Indicates debugging information for the OER border router learning process.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer border routes

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer border routes** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display debugging information for OER-controlled or monitored routes on the local border router, use the **debug oer border routes** command in privileged EXEC mode. To stop the display of debug event information, use the **no** form of this command.

debug oer border routes {**bgp** | **eigrp** [**detail**] | **piro** [**detail**] | **static**}
no debug oer border routes {**bgp** | **eigrp** | **static** | **piro**}

Syntax Description

bgp	Displays debugging information for BGP routes.
eigrp	Displays debugging information for EIGRP routes.
detail	(Optional) Displays detailed debugging information. This keyword applies only to EIGRP or PIRO routes.
static	Displays debugging information for static routes.
piro	Displays debugging information for Protocol Independent Route Optimization (PIRO) routes.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(24)T	This command was modified. The **piro** keyword was added to support the Protocol Independent Route Optimization (PIRO) feature.
15.0(1)M	This command was modified. The **eigrp** keyword was added to support EIGRP route control.
12.2(33)SRE	This command was modified. The **eigrp** keyword was added to support EIGRP route control and the **piro** keyword was added to support the PIRO feature.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer border routes** command is entered on a border router. This command is used to display the debugging information about OER-controlled or monitored routes on the local border router.

In Cisco IOS Release 12.4(24)T, 12.2(33)SRE, and later releases, PIRO introduced the ability for OER to search for a parent route--an exact matching route, or a less specific route--in any IP Routing Information Base (RIB). If a parent route for the traffic class exists in the RIB, policy-based routing is used to control the prefix.

In Cisco IOS Release 15.0(1)M, 12.2(33)SRE, and later releases, EIGRP route control introduced the ability for OER to search for a parent route--an exact matching route, or a less specific route--in the EIGRP routing table. If a parent route for the traffic class exists in the EIGRP routing table, temporary EIGRP routes are injected and identified by adding a configurable extended community tag value.

Examples

The following example enables the display of active-probe debug information on a border router:

```
Router# debug oer border routes
 bgp
*May  4 22:35:53.239: OER BGP: Control exact prefix 10.1.5.0/24
*May  4 22:35:53.239: OER BGP: Walking the BGP table for 10.1.5.0/24
*May  4 22:35:53.239: OER BGP: Path for 10.1.5.0/24 is now under OER control
*May  4 22:35:53.239: OER BGP: Setting prefix 10.1.5.0/24 as OER net#
```

The table below describes the significant fields shown in the display.

Table 46: debug oer border routes Field Descriptions

Field	Description
OER BGP:	Indicates debugging information for OER-controlled BGP routes.
OER STATIC:	Indicates debugging information for OER-controlled Static routes. (Not displayed in the example output.)

The following example enables the display of detailed debugging information for PIRO routes and shows that the parent route for the prefix 10.1.1.0 is found in the RIB and a route map is created to control the application. Note that detailed border PBR debugging is also active. This example requires Cisco IOS Release 12.4(24)T, 12.2(33)SRE, or a later release.

```
Router# debug oer border routes piro detail
Feb 21 00:20:44.431: PIRO: Now calling ip_get_route
Feb 21 00:20:44.431: PFR PIRO: Parent lookup found parent 10.1.1.0, mask 255.255.255.0,
nexthop 10.1.1.0 for network 10.1.1.0/24
...
Feb 21 00:22:46.771: PFR PIRO: Parent lookup found parent 10.1.1.0, mask 255.255.255.0,
nexthop 10.1.1.0 for network 10.1.1.0/24
Feb 21 00:22:46.771: PFR PIRO: Control Route, 10.1.1.0/24, NH 0.0.0.0, IF Ethernet4/2
Feb 21 00:22:46.771: PIRO: Now calling ip_get_route
Feb 21 00:22:46.771: PIRO: Now calling ip_get_route
Feb 21 00:22:46.771: PFR PIRO: Parent lookup found parent 10.1.1.0, mask 255.255.255.0,
nexthop 10.1.1.0 for network 10.1.1.0/24
Feb 21 00:22:46.771: OER BR PBR(det): control app: 10.1.1.0/24, nh 0.0.0.0, if
Ethernet4/2,ip prot 256, dst opr 0, src opr 0, 0 0 0 0, src net 0.0.0.0/0, dscp 0/0
Feb 21 00:22:46.771: OER BR PBR(det): Create rmap 6468E488
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) T 10.1.1.0/24 EVENT Track
start
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) N 10.1.1.0/24 Adding track
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) N 10.1.1.0/24 QP Schedule
query
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) T 10.1.1.0/24 EVENT Query
found route
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) N 10.1.1.0/24 Adding route
```

```
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) R 10.1.1.0/24  d=0 p=0 ->
Updating
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) R 10.1.1.0/24  d=110 p=1 ->
Et4/2 40.40.40.2 40 Notifying
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: Adding to client notification queue
Feb 21 00:22:46.775: PfR-RIB RIB_RWATCH: (default:ipv4:base) W 10.1.1.0/24 c=0x15 Client
notified reachable
Feb 21 00:22:46.779: PFR PIRO: Route update rwinfo 680C8E14, network 10.1.1.0, mask_len 24
event Route Up
Feb 21 00:22:46.779: OER BR PBR(det): PIRO Path change notify for prefix:10.1.1.0,
masklen:24, reason:1
```

The table below describes the significant fields shown in the display.

Table 47: debug oer border routes Field Descriptions

Field	Description
PFR PIRO	Indicates debugging information for Performance Routing-controlled PIRO activities.
OER BR PBR	Indicates debugging information about policy-based routing activities on the border router.
PfR-RIB RIB_RWATCH	Indicates debugging information about RIB activities.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer border traceroute reporting

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer border traceroute reporting** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display debugging information for traceroute probes on the local border router, use the **debug oer border traceroute reporting** command in privileged EXEC mode. To stop the display of debug event information, use the **no** form of this command.

debug oer border traceroute reporting [**detail**]
no debug oer border traceroute reporting [**detail**]

Syntax Description

detail	(Optional) Displays detailed traceroute debug information.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer border traceroute reporting**command is entered on a border router. This command is used to display the debugging information about traceroute probes sourced on the local border router.

Examples

The following example enables the display of active-probe debug information on a border router:

```
Router# debug oer border traceroute reporting

May 19 03:46:23.807: OER BR TRACE(det): Received start message: msg1 458776,
msg2 1677787648, if index 19, host addr 100.1.2.1, flags 1, max ttl 30,
protocol 17, probe delay 0
May 19 03:46:26.811: OER BR TRACE(det): Result msg1 458776,
msg2 1677787648 num hops 30 sent May 19 03:47:20.919: OER BR TRACE(det):
Received start message: msg1 524312, msg2 1677787648, if index 2,
host addr 100.1.2.1, flags 1, max ttl 30, protocol 17, probe delay 0
May 19 03:47:23.923: OER BR TRACE(det): Result msg1 524312,
msg2 1677787648 num hops 3 sent
```

The table below describes the significant fields shown in the display.

Table 48: debug oer border traceroute reporting Field Descriptions

Field	Description
OER BR TRACE:	Indicates border router debugging information for traceroute probes.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer cc

Note Effective with Cisco IOS Release 15.0(1)SY, the **debug oer cc** command is hidden. Although this command is still available in Cisco IOS software, the CLI interactive Help does not display it if you attempt to view it by entering a question mark at the command line. This command will be completely removed in a future release.

To display OER communication control debugging information for master controller and border router communication, use the **debug oer cc** command in privileged EXEC mode. To stop the display of OER debugging information, use the **no** form of this command.

debug oer cc [**detail**]
no debug oer cc [**detail**]

Syntax Description

detail	(Optional) Displays detailed information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.0(1)SY	This command was modified. This command was hidden.

Usage Guidelines

The **debug oer cc** command can be entered on a master controller on a border router. This command is used to display messages exchanged between the master controller and the border router. These messages include control commands, configuration commands, and monitoring information. Enabling this command will cause very detailed output to be displayed and can utilize a considerable amount of system resources. This command should be enabled with caution in a production network.

Examples

The following example enables the display of OER communication control debugging messages:

```
Router# debug oer cc
*May  4 23:03:22.527: OER CC: ipflow prefix reset received: 10.1.5.0/24
```

The table below describes the significant fields shown in the display.

Table 49: debug oer cc Field Descriptions

Field	Description
OER CC:	Indicates debugging information for OER communication messages.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master border

To display debugging information for OER border router events on an OER master controller, use the **debug oer master border**command in privileged EXEC mode. To stop border router event debugging, use the **no** form of this command.

debug oer master border [*ip-address*]
no debug oer master border

Syntax Description

ip-address	(Optional) Specifies the IP address of a border router.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master border** command is entered on a master controller. The output displays information related to the events or updates from one or more border routers.

Examples

The following example shows the status of 2 border routers. Both routers are up and operating normally.

```
Router# debug oer master border
OER Master Border Router debugging is on
Router#
1d05h: OER MC BR 10.4.9.7: BR I/F update, status UP, line 1 index 1, tx bw 10000
0, rx bw 100000, time, tx ld 0, rx ld 0, rx rate 0 rx bytes 3496553, tx rate 0,
tx bytes 5016033
1d05h: OER MC BR 10.4.9.7: BR I/F update, status UP, line 1 index 2, tx bw 10000
0, rx bw 100000, time, tx ld 0, rx ld 0, rx rate 0 rx bytes 710149, tx rate 0, t
x bytes 1028907
1d05h: OER MC BR 10.4.9.6: BR I/F update, status UP, line 1 index 2, tx bw 10000
0, rx bw 100000, time, tx ld 0, rx ld 0, rx rate 0 rx bytes 743298, tx rate 0, t
x bytes 1027912
1d05h: OER MC BR 10.4.9.6: BR I/F update, status UP, line 1 index 1, tx bw 10000
0, rx bw 100000, time, tx ld 0, rx ld 0, rx rate 0 rx bytes 3491383, tx rate 0,
tx bytes 5013993
```

The table below describes the significant fields shown in the display.

Table 50: debug oer master border Field Descriptions

Field	Description
OER MC BR ip-address:	Indicates debugging information for a border router process. The ip-address identifies the border router.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master collector

To display data collection debugging information for OER monitored prefixes, use the **debug oer master collector**command in privileged EXEC mode. To disable the display of this debugging information, use the **no** form of this command.

debug oer master collector {**active-probes** [**detail** [**trace**]] | **netflow**}
no debug oer master collector {**active-probes** [**detail** [**trace**]] | **netflow**}

Syntax Description

active-probes	Displays aggregate active probe results for a given prefix on all border routers that are executing the active probe.
detail	(Optional) Displays the active probe results from each target for a given prefix on all border routers that are executing the active probe.
trace	(Optional) Displays aggregate active probe results and historical statistics for a given prefix on all border routers that are executing the active probe.
netflow	Displays information about the passive (NetFlow) measurements received by the master controller for prefixes monitored from the border router.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master collector** command is entered on a master controller. The output displays data collection information for monitored prefixes.

Examples

debug oer master collector active-probes Example

The following example displays aggregate active probe results for the 10.1.0.0/16 prefix on all border routers that are configured to execute this active probe:

```
Router# debug oer master collector active-probes

*May  4 22:34:58.221: OER MC APC: Probe Statistics Gathered for prefix 10.1.0.0/16 on all
  exits,notifying the PDP
*May  4 22:34:58.221: OER MC APC: Summary Exit Data (pfx 10.1.0.0/16, bdr 10.2.2.2, if 13,

  nxtHop Default):savg delay 13, lavg delay 14, sinits 25, scompletes 25
*May  4 22:34:58.221: OER MC APC: Summary Prefix Data: (pfx 10.1.0.0/16) sloss 0, lloss 0,

  sunreach 25, lunreach 25, savg raw delay 15, lavg raw delay 15, sinits 6561,
  scompletes 6536, linits 6561, lcompletes 6536
*May  4 22:34:58.221: OER MC APC: Active OOP check done
```

The table below describes the significant fields shown in the display.

Table 51: debug oer master collector active-probes Field Descriptions

Field	Description
OER MC APC:	Indicates debugging information for active probes from the r OER master collector.

debug oer master collector active-probes detail Example

The following example displays aggregate active probe results from each target for the 10.1.0.0/16 prefix on all border routers that are configured to execute this active probe:

```
Router# debug oer master collector active-probes detail

*May  4 22:36:21.945: OER MC APC: Rtrv Probe Stats: BR 10.2.2.2, Type echo,
  Tgt 10.1.1.1,TgtPt 0, Src Default, SrcPt 0, NxtHp Default, Ndx 13
*May  4 22:36:22.001: OER MC APC: Remote stats received: BR 10.2.2.2, Type
  echo, Tgt 10.15.1, TgtPt 0, Src Default, SrcPt 0, NxtHp Default, Ndx 13
*May  4 22:36:22.313: OER MC APC: Perf data point (pfx 10.1.0.0/16, bdr
  10.2.2.2, if 13, xtHop Default):  avg delay 20, loss 0, unreach 0,
  initiations 2, completions 2, delay sum40, ldelay max 20, ldelay min 12
*May  4 22:36:22.313: OER MC APC: Perf data point (pfx 10.1.0.0/16, bdr
  10.2.2.2, if 13, xtHop Default):  avg delay 20, loss 0, unreach 0,
  initiations 2, completions 2, delay sum40, ldelay max 20, ldelay min 12
*May  4 22:36:22.313: OER MC APC: Probe Statistics Gathered for prefix
  10.1.0.0/16 on al exits, notifying the PDP
*May  4 22:36:22.313: OER MC APC: Active OOP check done
```

The table below describes the significant fields shown in the display.

Table 52: debug oer master collector active-probes detail Field Descriptions

Field	Description
OER MC APC:	Indicates debugging information for active probes from the r OER master collector.

debug oer master collector active-probes detail trace Example

The following example displays aggregate active probe results and historical statistics from each target for the 10.1.0.0/16 prefix on all border routers that are configured to execute this active probe:

```
Router# debug oer master collector active-probes detail trace

*May  4 22:40:33.845: OER MC APC: Rtrv Probe Stats: BR 10.2.2.2, Type echo,
  Tgt 10.1.5.1, TgtPt 0, Src Default, SrcPt 0, NxtHp Default, Ndx 13
*May  4 22:40:33.885: OER MC APC: Remote stats received: BR 10.2.2.2, Type
  echo, Tgt 10.1.5.1, TgtPt 0, Src Default, SrcPt 0, NxtHp Default, Ndx 13
*May  4 22:40:34.197: OER MC APC: Remote stats received: BR 10.2.2.2, Type
  echo, Tgt 10.1.2.1, TgtPt 0, Src Default, SrcPt 0, NxtHp Default, Ndx 13
*May  4 22:40:34.197: OER MC APC: Updating Probe (Type echo Tgt 10.1.2.1
  TgtPt 0) Total Completes 1306, Total Attempts 1318
*May  4 22:40:34.197: OER MC APC: All stats gathered for pfx 10.1.0.0/16
  Accumulating Stats
*May  4 22:40:34.197: OER MC APC: Updating Curr Exit Ref (pfx 10.1.0.0/16,
```

```
  bdr 10.2.2.2, if 13, nxtHop Default) savg delay 17, lavg delay 14, savg loss
  0, lavg loss 0, savg unreach 0, lavg unreach 0
*May  4 22:40:34.197: OER MC APC: Probe Statistics Gathered for prefix
  10.1.0.0/16 on all exits, notifying the PDP
*May  4 22:40:34.197: OER MC APC: Active OOP check done
```

The table below describes the significant fields shown in the display.

Table 53: debug oer master collector active-probes detail trace Field Descriptions

Field	Description
OER MC APC:	Indicates debugging information for active probes from the r OER master collector.

debug oer master collector netflow Example

The following example displays passive monitoring results for the 10.1.5.0/24 prefix:

```
Router# debug oer master collector netflow

*May  4 22:31:45.739: OER MC NFC: Rcvd  egress update from BR 10.1.1.2
  prefix 10.1.5.0/24  Interval 75688  delay_sum 0  samples 0  bytes 20362  pkts 505
  flows 359  pktloss 1 unreach 0
*May  4 22:31:45.739: OER MC NFC: Updating exit_ref; BR 10.1.1.2 i/f Et1/0,
  s_avg_delay 655, l_avg_delay 655, s_avg_pkt_loss 328, l_avg_pkt_loss 328,
  s_avg_flow_unreach 513, l_avg_flow_unreach 513
*May  4 22:32:07.007: OER MC NFC: Rcvd ingress update from BR 10.1.1.3
  prefix 10.1.5.0/24  Interval 75172  delay_sum 42328  samples 77  bytes 22040
  pkts 551  flows 310  pktloss 0 unreach 0
```

The table below describes the significant fields shown in the display.

Table 54: debug oer master collector netflow Field Descriptions

Field	Description
OER MC NFC:	Indicates debugging information for the OER master collector from passive monitoring (NetFlow).

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master cost-minimization

To display debugging information for cost-based optimization policies, use the **debug oer master cost-minimization** command in privileged EXEC mode. To disable the display of this debugging information, use the **no** form of this command.

debug oer master cost-minimization [**detail**]
no debug oer master cost-minimization [**detail**]

Syntax Description

detail	(Optional) Displays detailed information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master cost-minimization** command is entered on a master controller. The output displays debugging information for cost-minimization policies.

Examples

The following example displays detailed cost optimization policy debug information:

```
Router# debug oer master cost-minimization detail
OER Master cost-minimization Detail debugging is on
*May 14 00:38:48.839: OER MC COST: Momentary target utilization for exit 10.1.1.2 i/f
Ethernet1/0 nickname ISP1 is 7500 kbps, time_left 52889 secs, cumulative 16 kb, rollup
period 84000 secs, rollup target 6000 kbps, bw_capacity 10000 kbps
*May 14 00:38:48.839: OER MC COST: Cost OOP check for border 10.1.1.2, current util: 0
target util: 7500 kbps
*May 14 00:39:00.199: OER MC COST: ISP1 calc separate rollup ended at 55 ingress Kbps
*May 14 00:39:00.199: OER MC COST: ISP1 calc separate rollup ended at 55 egress bytes
*May 14 00:39:00.199: OER MC COST: Target utilization for nickname ISP1 set to 6000,
rollups elapsed 4, rollups left 24
*May 14 00:39:00.271: OER MC COST: Momentary target utilization for exit 10.1.1.2 i/f
Ethernet1/0 nickname ISP1 is 7500 kbps, time_left 52878 secs, cumulative 0 kb, rollup
period 84000 secs, rollup target 6000 kbps, bw_capacity 10000 kbps
*May 14 00:39:00.271: OER MC COST: Cost OOP check for border 10.1.1.2, current util: 0
target util: 7500 kbps
```

The table below describes the significant fields shown in the display.

Table 55: debug oer master cost-minimization detail Field Descriptions

Field	Description
OER MC COST:	Indicates debugging information for cost-based optimization on the master controller.

Related Commands

Command	Description
cost-minimization	Configures cost-based optimization policies on a master controller.
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.
show oer master cost-minimization	Displays the status of cost-based optimization policies.

debug oer master exit

To display debug event information for OER managed exits, use the **debug oer master exit**command in privileged EXEC mode. To stop the display of debug event information, use the **no** form of this command.

debug oer master exit [**detail**]
no debug oer master exit [**detail**]

Syntax Description

detail	Displays detailed OER managed exit information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master exit** command is entered on a master controller. This command is used to display debugging information for master controller exit selection processes.

Examples

The following example shows output form the **debug oer master exit** command, entered with the **detail**keyword:

```
Router# debug oer master exit
 detail
*May  4 11:26:51.539: OER MC EXIT: 10.1.1.1, intf Fa4/0 INPOLICY
*May  4 11:26:52.195: OER MC EXIT: 10.2.2.3, intf Se2/0 INPOLICY
*May  4 11:26:55.515: OER MC EXIT: 10.1.1.2, intf Se5/0 INPOLICY
*May  4 11:29:14.987: OER MC EXIT: 7 kbps should be moved from 10.1.1.1, intf Fa4/0
*May  4 11:29:35.467: OER MC EXIT: 10.1.1.1, intf Fa4/0 in holddown state so skip OOP check

*May  4 11:29:35.831: OER MC EXIT: 10.2.2.3, intf Se2/0 in holddown state so skip OOP check

*May  4 11:29:39.455: OER MC EXIT: 10.1.1.2, intf Se5/0 in holddown state so skip OOP check
```

The table below describes the significant fields shown in the display.

Table 56: debug oer master exit detail Field Descriptions

Field	Description
OER MC EXIT:	Indicates OER master controller exit event.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master learn

To display debug information for OER master controller learning events, use the **debug oer master learn** command in privileged EXEC mode. To stop the display of debug information, use the **no** form of this command.

debug oer master learn
no debug oer master learn

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master learn** command is entered on a master controller. This command is used to display debugging information for master controller learning events.

Examples

The following example shows output from the **debug oer master learn** command. The output an shows OER Top Talker debug events. The master controller is enabling prefix learning for new border router process:

```
Router# debug oer master learn
06:13:43: OER MC LEARN: Enable type 3, state 0
06:13:43: OER MC LEARN: OER TTC: State change, new RETRY, old DISABLED, reason TT start
06:13:43: OER MC LEARN: OER TTC: State change, new RETRY, old DISABLED, reason TT start
request
06:13:43: OER MC LEARN: OER TTC: State change, new RETRY, old DISABLED, reason T
T start request
06:14:13: OER MC LEARN: TTC Retry timer expired
06:14:13: OER MC LEARN: OER TTC: State change, new STARTED, old RETRY, reason At
 least one BR started
06:14:13: %OER_MC-5-NOTICE: Prefix Learning STARTED
06:14:13: OER MC LEARN: MC received BR TT status as enabled
06:14:13: OER MC LEARN: MC received BR TT status as enabled
06:19:14: OER MC LEARN: OER TTC: State change, new WRITING DATA, old STARTED, reason
Updating DB
06:19:14: OER MC LEARN: OER TTC: State change, new SLEEP, old WRITING DATA, reason
Sleep state
```

The table below describes the significant fields shown in the display.

Table 57: debug oer master learn Field Descriptions

Field	Description
OER MC LEARN:	Indicates OER master controller learning events.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master prefix

To display debug events related to prefix processing on an OER master controller, use the **debug oer master prefix**command in privileged EXEC mode. To disable the display of debug information, use the **no** form of this command.

debug oer master prefix [{*prefix* | **appl**}] [**detail**]
no debug oer master prefix [{*prefix* | **appl**}] [**detail**]

Syntax Description

prefix	(Optional) Specifies a single prefix or prefix range. The prefix address and mask are entered with this argument.
appl	(Optional) Displays information about prefixes used by applications monitored and controlled by an OER master controller.
detail	(Optional) Displays detailed OER prefix processing information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master prefix** command is entered on a master controller. This command displays debugging information related to prefix monitoring and processing.

Examples

The following example shows the master controller searching for the target of an active probe after the target has become unreachable.

```
Router# debug oer master prefix

OER Master Prefix debugging is on
06:01:28: OER MC PFX 10.4.9.0/24: APC last target deleted for prefix, no targets
 left assigned and running
06:01:38: OER MC PFX 10.4.9.0/24: APC Attempting to probe all exits
06:02:59: OER MC PFX 10.4.9.0/24: APC last target deleted for prefix, no targets
 left assigned and running
06:03:08: OER MC PFX 10.4.9.0/24: APC Attempting to probe all exits
06:04:29: OER MC PFX 10.4.9.0/24: APC last target deleted for prefix, no targets
 left assigned and running
06:04:39: OER MC PFX 10.4.9.0/24: APC Attempting to probe all exits
06:05:59: OER MC PFX 10.4.9.0/24: APC last target deleted for prefix, no targets
 left assigned and running
06:06:09: OER MC PFX 10.4.9.0/24: APC Attempting to probe all exits
```

The table below describes the significant fields shown in the display.

Table 58: debug oer master prefix Field Descriptions

Field	Description
OER MC PFX ip-address:	Indicates debugging information for OER monitored prefixes. The ip-address identifies the prefix.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master prefix-list

To display debug events related to prefix-list processing on an OER master controller, use the **debug oer master prefix-list**command in privileged EXEC mode. To disable the display of debug information, use the **no** form of this command.

debug oer master prefix-list *list-name* [**detail**]
no debug oer master prefix-list *list-name*

Syntax Description

list-name	Specifies a single prefix or prefix range. The prefix address and mask are entered with this argument.
detail	(Optional) Displays detailed OER prefix-list processing information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(11)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master prefix-list**command is entered on a master controller. This command displays debugging information related to prefix-list processing.

Examples

The following example shows output from the **debug oer master prefix-list**command.

```
Router# debug oer master prefix-list

23:02:16.283: OER MC PFX 10.1.5.0/24: Check PASS REL loss: loss 0, policy 10%, notify TRUE
23:02:16.283: OER MC PFX 10.1.5.0/24: Passive REL loss in-policy
23:02:16.283: OER MC PFX 10.1.5.0/24: Check PASS REL delay: delay 124, policy 50%, notify
TRUE
23:02:16.283: OER MC PFX 10.1.5.0/24: Passive REL delay in policy
23:02:16.283: OER MC PFX 10.1.5.0/24: Prefix not OOP
23:02:16.283: OER MC PFX 10.1.5.0/24: Check PASS REL unreachable: unreachable 0, policy
50%, notify TRUE
23:02:16.283: OER MC PFX 10.1.5.0/24: Passive REL unreachable in-policy
23:02:16.283: OER MC PFX 10.1.5.0/24: Check PASS REL loss: loss 0, policy 10%, notify TRUE
23:02:16.283: OER MC PFX 10.1.5.0/24: Passive REL loss in policy
```

The table below describes the significant fields shown in the display.

Table 59: debug oer master prefix-list Field Descriptions

Field	Description
OER MC PFX ip-address:	Indicates debugging information for OER monitored prefixes. The ip-address identifies the prefix.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master process

To display debug information about the OER master controller process, use the **debug oer master process** command in privileged EXEC mode. To stop displaying debug information, use the **no** form of this command.

debug oer master process
no debug oer master process

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

The **debug oer master process** command is entered on a master controller.

Examples

The following sample debug output for a master controller process:

```
Router# debug oer master process
01:12:00: OER MC PROCESS: Main msg type 15, ptr 0, value 0
```

The table below describes the significant fields shown in the display.

Table 60: debug oer master process Field Descriptions

Field	Description
OER MC PROCESS:	Indicates a master controller master process debugging message.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug oer master traceroute reporting

To display debug information about traceroute probes, use the **debug oer master traceroute reporting** command in privileged EXEC mode. To stop displaying debug information, use the **no** form of this command.

debug oer master traceroute reporting [**detail**]
no debug oer master traceroute reporting [**detail**]

Syntax Description

detail	(Optional) Displays detailed information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

The **debug oer master traceroute reporting** command is entered on a master controller. This command is used to display traceroute events on a master controller.

Examples

The following sample debug output for a master controller process:

```
Router# debug oer master traceroute reporting detail
*May 12 18:55:14.239: OER MC TRACE: sent start message msg1 327704, msg2 167838976,
if index 2, host add 10.1.5.2, flags 1, max ttl 30, protocol 17
*May 12 18:55:16.003: OER MC TRACE: sent start message msg1 393240, msg2 167838976,
if index 2, host add 10.1.5.2, flags 1, max ttl 30, protocol 17
master#
*May 12 18:55:17.303: OER MC TRACE: Received result: msg_id1 327704, prefix 10.1.5.0/24,
hops 4, flags 1
*May 12 18:55:19.059: OER MC TRACE: Received result: msg_id1 393240, prefix 10.1.5.0/24,
hops 4, flags 1
```

The table below describes the significant fields shown in the display.

Table 61: debug oer master traceroute reporting detail Field Descriptions

Field	Description
OER MC PROCESS:	Indicates master controller debugging information for traceroute probes.

Related Commands

Command	Description
oer	Enables an OER process and configures a router as an OER border router or as an OER master controller.

debug ospfv3

To display debugging information for Open Shortest Path First version 3 (OSPF) for IPv4 and IPv6, use the **debug ospfv3**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ospfv3 [*process-id*] [*address-family*] [{**adj** | **ipsec** | **database-timer** | **flood** | **hello** | **lsa-generation** | **retransmission**}]
no debug ospfv3 [*process-id*] [*address-family*] [{**adj** | **ipsec** | **database-timer** | **flood** | **hello** | **lsa-generation** | **retransmission**}]

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.
adj	(Optional) Displays adjacency information.
ipsec	(Optional) Displays the interaction between OSPFv3 and IPSec, including creation and removal of policy definitions.
database-timer	(Optional) Displays database-timer information.
flood	(Optional) Displays flooding information.
hello	(Optional) Displays hello packet information.
l2api	(Optional) Enables layer 2 and layer 3 application program interface (API) debugging.
lsa-generation	(Optional) Displays link-state advertisement (LSA) generation information for all LSA types.
retransmission	(Optional) Displays retransmission information.

Command Default

Debugging of OSPFv3 is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Consult Cisco technical support before using this command.

Examples

The following example displays adjacency information for OSPFv3:

```
Device# debug ospfv3 adj
```

debug ospfv3 authentication

To display the debugging information for Open Shortest Path First version 3 (OSPF) for VPN routing and forwarding (VRF) authentication, use the

debug ospfv3 authentication command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ospfv3 [**pid**] [**vrf** {***** | *instance-name*] **authentication**
no debug ospfv3 [**vrf** {***** | *instance-name*] **authentication**

Syntax Description

pid	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
vrf	(Optional) The virtual routing and forwarding instance.
*****	Includes all VPN routing and forwarding instances.
instance-name	Name of a specific VPN routing and forwarding instance.

Command Modes

Privileged EXEC

Command History

Release	Modification
Cisco IOS XE Release 3.11S	This command was introduced.

Usage Guidelines

Consult Cisco technical support before using this command.

Examples

The following example displays the VRF authentication for OSPFv3:

```
Device# debug ospfv3 vrf * authentication

OSPFv3 Authentication events debugging is on
```

debug ospfv3 database-timer rate-limit

To display debugging information about the current wait-time used for shortest path first (SPF) scheduling, use the **debug ospfv3 database-timer rate-limit**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ospfv3 [*process-id*] [*address-family*] **database-timer rate-limit** [*acl-number*]
no debug ospfv3 [*process-id*] [*address-family*] **database-timer rate-limit**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.
acl-number	(Optional) Access list number.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Consult Cisco technical support before using this command.

Examples

The following example shows how to turn on debugging for SPF scheduling in OSPFv3 process 1:

```
Device# debug ospfv3 1 database-timer rate-limit
```

debug ospfv3 events

To display information on Open Shortest Path First version 3 (OSPFv3)-related events, such as designated router selection and shortest path first (SPF) calculation, use the **debug ospfv3 events** command in privileged EXEC com mand. To disable debugging output, use the **no** form of this command.

debug ospfv3 [*process-id*] [*address-family*] **events**
no debug ipv6 ospfv3 [*process-id*] [*address-family*] **events**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Consult Cisco technical support before using this command.

Examples

The following example displays information on OSPFv3-related events:

```
Device#
debug ospfv3 events
```

debug ospfv3 lsa-maxage

display debug messages about OSPFv3 LSA MaxAge events, use the **debug ospfv3 lsa-maxage** command in privileged EXEC mode. To disable the display of debug messages, use the **no** form of this command.

debug ospfv3 [*address-family*] **lsa-maxage** [*access-list-number*]

no debug ospfv3 [*address-family*] **lsa-maxage** [*access-list-number*]

Syntax Description

address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.
access-list-number	(Optional) Access list number. IP access lists are in the range 1 to 99.

Command Default

By default, debug messages about OSPFv3 LSA MaxAge events are not displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
Cisco IOS XE 17.1.1	Command introduced.

Usage Guidelines

Debug messages are displayed for every LSA for which a MaxAge event occurs. To limit the output, use an access list.

Example

```
router#debug ospfv3 lsa-maxage
OSPFv3 LSA maxage debugging is on for process 10, IPv4, Default vrf
OSPFv3 LSA maxage debugging is on for process 10, IPv6, Default vrf
```

debug ospfv3 lsdb

To display database modifications for Open Shortest Path First version 3 (OSPFv3), use the **debug ospfv3 lsdb** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ospfv3 [*process-id*] [*address-family*] **lsdb**
no debug ospfv3 [*process-id*] [*address-family*] **lsdb**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Consult Cisco technical support before using this command.

Examples

The following example displays database modification information for OSPFv3:

```
Device# debug ospfv3 lsdb
```

debug ospfv3 packet

To display information about each Open Shortest Path First version 3 (OSPFv3) packet received, use the **debug ospfv3 packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ospfv3 [*process-id*] [*address-family*] **packet**
no debug ospfv3 [*process-id*] [*address-family*] **packet**

Syntax Description

process-id	(Optional) Internal identification. The number used here is the number assigned administratively when enabling the OSPFv3 routing process and can be a value from 1 through 65535.
address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Consult Cisco technical support before using this command.

Examples

The following example displays information about each OSPFv3 packet received:

```
Router# debug ospfv3 packet
```

debug ospfv3 spf statistic

To display statistical information while running the shortest path first (SPF) algorithm, use the **debug ospfv3 spf statistic**command in privileged EXEC mode. To disable the debugging output, use the **no** form of this command.

debug ospfv3 [*address-family*] **spf statistic**
no debug ospfv3 [*address-family*] **spf statistic**

Syntax Description

address-family	(Optional) Enter **ipv6** for the IPv6 address family or **ipv4** for the IPv4 address family.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.1(3)S	This command was introduced.
Cisco IOS XE Release 3.4S	This command was integrated into Cisco IOS XE Release 3.4S.
15.2(1)T	This command was integrated into Cisco IOS Release 15.2(1)T.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

The **debug ospfv3 spf statistic** command displays the SPF calculation times in milliseconds, the node count, and a time stamp. Consult Cisco technical support before using this command.

Examples

The following example displays statistical information while running the SPF algorithm:

```
Router# debug ospfv3 spf statistics
```

Related Commands

Command	Description
debug ospfv3	Displays debugging information for the OSPFv3 feature.
debug ospfv3 events	Displays information on OSPFv3-related events.
debug ospfv3 packet	Displays information about each OSPFv3 packet received.

debug otv

To enable debugging of Overlay Transport Virtualization (OTV) and Intermediate System-to-Intermediate System (IS-IS) activities, use the **debug otv** command in privileged EXEC mode. To disable the debugging, use the **no** form of this command.

debug otv [{**adjacency** | **all** | **arp-nd** | **database** | **error** | **evc** | **event** | **ha** | **igp** | **isis** | **l2rib** | **l2rtgvpn** | **misc** | **multicast** | **overlay** | **packet** | **pim** | **state** | **tunnel** | **ui**}]
no debug otv [{**adjacency** | **all** | **arp-nd** | **database** | **error** | **evc** | **event** | **ha** | **igp** | **isis** | **l2rib** | **l2rtgvpn** | **misc** | **multicast** | **overlay** | **packet** | **pim** | **state** | **tunnel** | **ui**}]

Syntax Description

adjacency	Enables logging of adjacency-related events.
all	Enables logging of all debugging messages.
arp-nd	Enables logging of OTV database-related operations.
database	Enables logging of the Address Routing Protocol (ARP) suppression feature.
error	Enables logging of error debug messages.
evc	Enables logging of Ethernet Virtual Connections (EVC) interactions.
event	Enables logging of the event dispatcher.
ha	Enables logging of high availability (HA) events.
igp	Enables logging of OTV IS-IS events.
isis	Enables logging of IS-IS information.
l2rib	Enables logging of Layer 2 Routing Information Base (L2RIB) interactions.
l2rtgvpn	Enables logging of Layer 2 routing VPN manager.
misc	Enables logging of miscellaneous OTV debug messages.
multicast	Enables logging of multicast-related events.
overlay	Enables logging of overlay interface events.
packet	Enables logging of OTV packet forwarding activities.
pim	Enables logging of Protocol Independent Multicast (PIM) messages.
state	Enables logging of OTV state change events.
tunnel	Enables logging of tunnel interactions.
ui	Enables logging of OTV user interface (UI) events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following example shows how to enable logging of OTV activities:

```
Router# debug otv all

OTV APP all debugging is on
Router#
*Oct 27 13:53:45.155: OTV-APP-PKT-RX: Received packet on Overlay1 L3 dest 224.1.1.2 source
 209.165.201.20, L2 dest 0100.0cdf.dfdf source 0023.33cc.ebbc, linktype 25
*Oct 27 13:53:46.241: OTV-APP-PKT-RX: Received packet on Overlay1 L3 dest 224.1.1.2 source
 209.165.201.20, L2 dest 0100.0cdf.dfdf source 0015.17b9.c479, linktype 25
*Oct 27 13:53:46.824: OTV-APP-PKT-RX: Received packet on Overlay1 L3 dest 224.1.1.2 source
 209.165.201.20, L2 dest 0100.0cdf.dfdf source 0023.33cc.ebbc, linktype 25
*Oct 27 13:53:49.166: OTV-APP-PKT-RX: Received packet on Overlay1 L3 dest 224.1.1.2 source
 209.165.201.20, L2 dest 0100.0cdf.dfdf source 0015.17b9.c479, linktype 25
*Oct 27 13:53:50.055: OTV-APP-PKT-RX: Received packet on Overlay1 L3 dest 224.1.1.2 source
 209.165.201.20, L2 dest 0100.0cdf.dfdf source 0023.33cc.ebbc, linktype 25
*Oct 27 13:53:50.085: OTV-APP-PKT-TX: Overlay 1 process switching packet to 224.1.1.2
```

Related Commands

Command	Description
interface overlay	Creates an OTV overlay interface.
show otv	Displays OTV information.

debug otv isis

To enable debugging of Overlay Transport Virtualization (OTV) Intermediate System-to-Intermediate System (IS-IS) activities, use the **debug otv isis** command in privileged EXEC mode. To disable the debugging, use the **no** form of this command.

debug otv isis [**overlay** *interface*][**site**]{**adj-packets** *interface-type interface-number* | **aed** | **authentication information** | **checksum-errors** | **common event** | **local-updates** | **nsf**[{**cisco** | **detail** | **ietf**}] | **protocol-errors** | **rib**[**redistribution**][{**mac** | **multicast**[**mapping**]}] | **snp-packets** | **update-packets** | **vlan-database**}
no debug otv isis [**overlay** *interface*][**site**]{**adj-packets** *interface-type interface-number* | **aed** | **authentication information** | **checksum-errors** | **common event** | **local-updates** | **nsf**[{**cisco** | **detail** | **ietf**}] | **protocol-errors** | **rib**[**redistribution**][{**mac** | **multicast**[**mapping**]}] | **snp-packets** | **update-packets** | **vlan-database**}

Syntax Description

overlay *overlay-interface*	(Optional) Enables debugging of the specified overlay interface. The range is from 0 to 512.
site	(Optional) Enables logging of the IS-IS Layer 2 site process.
adj-packets	Enables logging of adjacency packets.
interface-type	Type of interface.
interface-number	Port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when added to a system; they can be displayed with the **show interfaces** command.
aed	Enables logging of authoritative edge device (AED) information.
authentication information	Enables logging of packet authentication information.
checksum-errors	Enables logging of link-state packet (LSP) checksum errors.
common event	Enables logging of common IS-IS events.
local-updates	Enables logging of local update packets.
nsf	Enables logging of IS-IS nonstop forwarding (NSF) information.
cisco	(Optional) Enables logging of only Cisco NSF information.
detail	(Optional) Enables logging of detailed NSF information.
ietf	(Optional) Enables logging of only IETF NSF information.
protocol-errors	Enables logging of LSP protocol errors.
rib	Enables logging of local Routing Information Base (RIB) events.
redistribution	(Optional) Enables logging of redistribution RIB events.
mac	(Optional) Enables logging of Layer 2 MAC RIB events.

multicast	(Optional) Enables logging of Layer 2 multicast RIB events.
mapping	(Optional) Enables logging of Layer 2 multicast mapping RIB events.
snp-packets	Enables logging of complete sequence number protocol data units (PDUs) (CSNP)/partial sequence number PDUs (PSNPs).
update-packets	Enables logging of update packets.
vlan-database	Enables logging of information about the VLAN database.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following is sample output from the **debug otv isis aed** command:

```
Router# debug otv isis aed

*Nov 11 22:16:21.309: ISIS-AEDInfo (Overlay1): Neighbor AABB.CC00.0300 not found in osn
list
*Nov 11 22:16:21.309: ISIS-AEDInfo (Overlay1): Neighbor AABB.CC00.0300 not found in osn
list
*Nov 11 22:16:21.309: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0300 in
site 0000.0000.0013
*Nov 11 22:16:21.309: ISIS-AEDInfo (Overlay1): Local AED enabled for isis
*Nov 11 22:16:21.309: ISIS-AEDInfo (Overlay1): adding neighbor AABB.CC00.0100 to osn list
*Nov 11 22:16:21.309: ISIS-AEDInfo (Overlay1): Adding site neighbor AABB.CC00.0100 to osn
*Nov 11 22:16:22.309: ISIS-AEDInfo (Overlay1): Neighbor AABB.CC00.0300 not found in osn
list
*Nov 11 22:16:43.182: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0300 in
site 0000.0000.0013
*Nov 11 22:16:43.182: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0300 in
site 0000.0000.0013
*Nov 11 22:16:43.182: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0300 in
site 0000.0000.0013
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): adding neighbor AABB.CC00.0200 to osn list
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): Adding site neighbor AABB.CC00.0200 to osn
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0200 in
site 0000.0000.0000
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): Removing overlay/all neighbor AABB.CC00.0200
 from osn
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): removed neighbor AABB.CC00.0200 from osn
list
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0300 in
site 0000.0000.0013
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): Found overlay neighbor AABB.CC00.0300 in
site 0000.0000.0013
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): adding neighbor AABB.CC00.0200 to osn list
*Nov 11 22:16:45.327: ISIS-AEDInfo (Overlay1): Adding overlay neighbor AABB.CC00.0200 to
osn
*Nov 11 22:16:48.144: ISIS-AEDInfo (Overlay1): Neighbor AABB.CC00.0200 already in osn list
*Nov 11 22:16:48.144: ISIS-AEDInfo (Overlay1): Adding site neighbor AABB.CC00.0200 to osn
```

Related Commands

Command	Description
show otv isis	Displays the IS-IS status and configuration.

debug packet

To display per-packet debugging output, use the **debugpacket**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug packet [{**interface** *number* [**vcd** *vcd-number*] | **vc** *vpi*/*vcivc-name*}]
no debug packet [{**interface** *number* [**vcd** *vcd-number*] | **vc** *vpi*/*vcivc-name*}]

Syntax Description

interface *number*	(Optional) interface or subinterface number.
vcd *vcd-number*	(Optional) Number of the virtual circuit designator (VCD).
vc *vpi* / *vci*	(Optional) Virtual path identifier (VPI) and virtual channel identifier (VCI) numbers of the VC.
vc-name	(Optional) Name of the PVC or SVC.

Command Default

Debugging for packets is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
9.21	This command was introduced.
12.2(13)T	Support for Apollo Domain and Banyan VINES was removed.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **debugpacket** command displays all process-level packets for both outbound and inbound packets. This command is useful for determining whether packets are being received and sent correctly. The output reports information online when a packet is received or a transmission is attempted.

For sent packets, the information is displayed only after the protocol data unit (PDU) is entirely encapsulated and a next hop VC is found. If information is not displayed, the address translation probably failed during encapsulation. When a next hop VC is found, the packet is displayed exactly as it will be presented on the wire. Having a display indicates that the packets are properly encapsulated for transmission.

For received packets, information is displayed for all incoming frames. The display can show whether the sending station properly encapsulates the frames. Because all incoming frames are displayed, this information is useful when performing back-to-back testing and corrupted frames cannot be dropped by an intermediary switch.

The **debugpacket** command also displays the initial bytes of the actual PDU in hexadecimal. This information can be decoded only by qualified support or engineering personnel.

Caution Because the **debugpacket** command generates a substantial amount of output for every packet processed, use it only when traffic on the network is low so other activity on the system is not adversely affected.

Examples

The following is sample output from the **debugpacket** command:

```
Router# debug packet
2/0.5(I): VCD:0x9 VCI:0x23 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800 Length0x70
4500 002E 0000 0000 0209 92ED 836C A26E FFFF FFFF 1108 006D 0001 0000 0000
A5CC 6CA2 0000 000A 0000 6411 76FF 0100 6C08 00FF FFFF 0003 E805 DCFF 0105
```

The following table describes the significant fields shown in the display.

Table 62: debug packet Field Descriptions

Field	Description
2/0.5	Indicates the subinterface that generated this packet.
(I)	Indicates a receive packet. (O) indicates an output packet.
VCD: 0x*n*	Indicates the virtual circuit associated with this packet, where *n* is some value.
DM: 0x*nnnn*	Indicates the descriptor mode bits on output only, where *nnnn* is a hexadecimal value.
TYPE:*n*	Displays the encapsulation type for this packet.
Length:*n*	Displays the total length of the packet including the headers.

The following two lines of output are the binary data, which are the contents of the protocol data unit (PDU) before encapsulation:

```
4500 002E 0000 0000 0209 92ED 836C A26E FFFF FFFF 1108 006D 0001 0000 0000
A5CC 6CA2 0000 000A 0000 6411 76FF 0100 6C08 00FF FFFF 0003 E805 DCFF 0105
```

The following is sample output from the **debugpacket** command:

```
Router# debug packet
Ethernet0: Unknown ARPA, src 0000.0c00.6fa4, dst ffff.ffff.ffff, type 0x0a0
data 00000c00f23a00000c00ab45, len 60
Serial3: Unknown HDLC, size 64, type 0xaaaa, flags 0x0F00
Serial2: Unknown PPP, size 128
Serial7: Unknown FRAME-RELAY, size 174, type 0x5865, DLCI 7a
Serial0: compressed TCP/IP packet dropped
```

The following table describes the significant fields shown in the display.

Table 63: debug packet Field Descriptions

Field	Description
Ethernet0	Name of the Ethernet interface that received the packet.
Unknown	Network could not classify this packet. Examples include packets with unknown link types.

Field	Description
ARPA	Packet uses ARPA-style encapsulation. Possible encapsulation styles vary depending on the media command mode (MCM) and encapsulation style. **Ethernet (MCM)** --*EncapsulationStyle:* • ARP • ETHERTALK • ISO1 • ISO3 • LLC2 • NOVELL-ETHER • SNAP
	FDDI (MCM) --*Encapsulation Style:* • ISO1 • ISO3 • LLC2 • SNAP
	Frame Relay --*EncapsulationStyle:* • BRIDGE • FRAME-RELAY

Field	Description
ARPA (continued)	**Serial (MCM)** --*EncapsulationStyle:* • BFEX25 • BRIDGE • DDN-X25 • DDNX25-DCE • ETHERTALK • FRAME-RELAY • HDLC • HDH • LAPB • LAPBDCE • MULTI-LAPB • PPP • SDLC-PRIMARY • SDLC-SECONDARY • SLIP • SMDS • STUN • X25 • X25-DCE
	Token Ring (MCM) --*EncapsulationStyle:* • 3COM-TR • ISO1 • ISO3 • MAC • LLC2 • NOVELL-TR • SNAP • VINES-TR
src 0000.0c00.6fa4	MAC address of the node generating the packet.

Field	Description
dst.ffff.ffff.ffff	MAC address of the destination node for the packet.
type 0x0a0	Packet type.
data...	First 12 bytes of the datagram following the MAC header.
len 60	Length of the message (in bytes) that the interface received from the wire.
size 64	Length of the message (in bytes) that the interface received from the wire. Equivalent to the len field.
flags 0x0F00	HDLC or PP flags field.
DLCI 7a	The DLCI number on Frame Relay.
compressed TCP/IP packet dropped	TCP header compression is enabled on an interface and the packet is not HDLC or X25.

debug packet-capture

To enable packet capture debugs, use the **debug packet-capture** command in privileged EXEC mode. To disable debugging packet capture, use the **no** form of this command.

debug packet-capture
no debug packet-capture

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS 12.2(33)SRE.

Examples

The following example shows output from a successful request when using the **debug packet-capture** command:

```
Router# debug packet-capture
Buffer Capture Infrastructure debugging is on
```

Related Commands

Command	Description
show monitor capture	Displays the contents of a capture buffer or a capture point.

debug pad

To display debugging messages for all packet assembler/disassembler (PAD) connections, use the **debug pad** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug pad
no debug pad

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0	This command was introduced in a release prior to Cisco IOS Release 12.0.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

Use the **debug pad** command to gather information to forward to the Cisco Technical Assistance Center (TAC) to assist in troubleshooting a problem that involves PAD connections.

The following example shows output of the **debug pad** and **debug x25 event** commands for an incoming PAD call destined for a terminal line. The incoming PAD call is rejected by the terminal line because the selected network closed user group (CUG) has not been subscribed to by the caller:

```
Router# debug pad
Router# debug x25 event
Serial1/1:X.25 I R1 Call (16) 8 lci 8
  From (7):2001534 To (9):200261150
  Facilities:(2)
    Closed User Group (basic):99
  Call User Data (4):0x01000000 (pad)
pad_svc_announce:destination matched 1
PAD:incoming call to 200261150 on line 130 CUD length 4
!PAD130:Incoming Call packet, Closed User Group (CUG) service protection, selected network
 CUG not subscribed
PAD:CUG service protection Cause:11 Diag:65
Serial1/1:X.25 O R1 Clear (5) 8 lci 8
  Cause 0, Diag 65 (DTE originated/Facility code not allowed)
Serial1/1:X.25 I R1 Clear Confirm (3) 8 lci 8
```

The following example shows the output of the **debug pad** command for an outgoing PAD call initiated from a terminal line with a subscribed CUG that bars outgoing access:

```
!PAD130:Outgoing Call packet, Closed User Group - CUG service validation, selected CUG !bars
 outgoing access
PAD130:Closing connection to .  In 0/0, out 0/0
```

debug piafs events

To check the debugging messages for Personal Handyphone Internet Access Forum Standard (PIAFS) calls, use the **debugpiafsevents** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug piafs events
no debug piafs events

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(8)T	This command was introduced on Cisco 803, Cisco 804, and Cisco 813 routers.

Usage Guidelines

The **debugpiafsevents**command provides debugging information for the PIAFS calls on the router, including the inband negotiation process.

Examples

The **debugpiafsevents** command was configured to provide the following information for PIAFS calls:

```
Router# debug piafs events
02:16:39:PIAFS events debugging is on
02:16:167516180371:PIAFS: RX <- CDAPI :cdapi_route_call Request
02:16:167517398148:PIAFS: RX <- CDAPI :CDAPI_MSG_CONNECT_IND
02:16:171798691839:PIAFS: TX -> CDAPI :CDAPI_MSG_SUBTYPE_ALERT_REQ
02:16:167503724545:PIAFS: TX -> CDAPI :CDAPI_MSG_CONNECT_RESP
02:16:167503765504:PIAFS: TX -> CDAPI :CDAPI_MSG_CONN_ACTIVE_REQ
02:16:167503724544:PIAFS: RX <- CDAPI :CDAPI_MSG_CONN_ACTIVE_IND
02:16:171798691839:PIAFS:Network allotted Channel :B1
02:16:167503765504:PIAFS:Enabling QMC in PIAFS mode for B1
02:16:171798691839:PIAFS:piafs_driver_enable_settings()
02:16:167503765504:PIAFS:The speed is :64
02:16:167503724544:PIAFS:Starting 64 kbps PIAFS Incoming
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:13 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:Updating conf resp num
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:1 RSN:1 CRSN:13 SISN:
255]
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:14 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:2 RSN:2 CRSN:13 SISN:
255]
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:15 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:3 RSN:3 CRSN:13 SISN:
255]
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:16 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:4 RSN:4 CRSN:13 SISN:
```

```
255]
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:17 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:5 RSN:5 CRSN:13 SISN:
255]
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:18 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:6 RSN:6 CRSN:13 SISN:
255]
02:16:39:PIAFS:RX <- NEGO_SYNC_REQUEST[GSN:19 RSN:1 CRSN:1 SISN:
255]
02:16:39:PIAFS:TX -> NEGO_SYNC_RECEPTION[GSN:7 RSN:7 CRSN:13 SISN:
255]
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:Piafs layer up & Main FSM set to DATA
02:16:39:PIAFS:Compression v42bis enabled
02:16:39:PIAFS:V42BIS:v42bis_init()
02:16:39:PIAFS:V42BIS:v42bis_init()
02:16:39:PIAFS:V42BIS:Negotiated Values for P1, P2 are - 4096 , 250
02:16:39:PIAFS:Incoming call invoking ISDN_CALL_CONNECT
02:16:39:%LINK-3-UPDOWN:Interface BRI0:1, changed state to up
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
```

```
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:RX <- CONTROL_REQUEST(comm parameter)[Seq No:0]
02:16:39:PIAFS:Rx Parameters:
02:16:39:PIAFS:  Data Protocol:Version 1
02:16:39:PIAFS:  Control Protocol:Version 1
02:16:39:PIAFS:  RTF value:9
02:16:39:PIAFS:  Compression:V.42bis
02:16:39:PIAFS:  Frame Length:80
02:16:39:PIAFS:  Frame Number:63
02:16:39:PIAFS:TX -> CONTROL_RECEPTION[0]
02:16:39:PIAFS:ACKed all the Rx control parameters
02:16:39:PIAFS:piafs_setmap() tx_map FFFFFFFF
02:16:39:PIAFS:piafs_setmap() rx_map 0
02:16:41:PIAFS:PPP:Autoselect sample 7E
02:16:41:PIAFS:PPP:Autoselect sample 7EFF
02:16:41:PIAFS:PPP:Autoselect sample 7EFF7D
02:16:41:PIAFS:PPP:Autoselect sample 7EFF7D23
02:16:41:PIAFS:piafs_setmap() tx_map FFFFFFFF
02:16:41:PIAFS:piafs_setmap() rx_map 0
02:16:42:PIAFS:piafs_setmap() tx_map A0000
02:16:42:PIAFS:piafs_setmap() rx_map 0
```

The following table describes the significant fields shown in the display.

Table 64: debug piafs events Field Descriptions

Field	Description
RX <- CDAPI :cdapi_route_call Request	The call distributor application programming interface (CDAPI) in the router receives an ISDN call request from the switch.
RX <- CDAPI :CDAPI_MSG_CONNECT_IND	The CDAPI in the router receives a connection indicator message from the switch.
TX -> CDAPI :CDAPI_MSG_SUBTYPE_ALERT_REQ	The CDAPI in the router transmits an alert request to the switch.

Field	Description
TX -> CDAPI :CDAPI_MSG_CONNECT_RESP	The CDAPI in the router transmits a connect response message to the switch.
TX -> CDAPI :CDAPI_MSG_CONN_ACTIVE_REQ	The CDAPI in the router transmits a connection active request to the switch.
RX <-CDAPI:CDAPI_MSG_CONN_ACTIVE_IND	The CDAPI in the router receives a connection active indicator from the switch.
Enabling QMC in PIAFS mode for B1	QMC (global multichannel parameters) are being enabled in PIAFS mode for the B1 channel.
piafs_driver_enable_settings()	The PIAFS driver is enabling the settings.
Starting 64 kbps PIAFS Incoming	The speed of the transmission in kbps. In this case, the speed is 64 kbps.
RX <- NEGO_SYNC_REQUEST[GSN: RSN: CRSN: SISN:]	The router receives a PIAFS negotiation synchronization request frame from the peer PIAFS device. The frame contains the following: general sequence number (GSN), reception sequence number (RSN), confirmation response sequence number (CRSN), and synchronization initiation sequence number (SISN).
Updating conf resp num	The confirmation response number is being updated.
TX -> NEGO_SYNC_RECEPTION[GSN: RSN: CRSN: SISN:]	The router transmits a PIAFS negotiation synchronization reception message to the peer PIAFS device. The message includes the GSN, RSN, CRSN, and SISN.
RX <- CONTROL_REQUEST	The router receives a PIAFS control request frame that includes communication parameters.
Rx Parameters	The communication parameters are as follows.
Data Protocol	The version of the data protocol.
Control Protocol	The version of the control protocol.
RTF value	Round-trip frame value.
Compression	The compression standard.
Frame Length	The length of the frame, in bytes.
Frame Number	The number of packets per frame.
TX -> CONTROL_RECEPTION	The router transmits a PIAFS control reception frame.
ACKed all the Rx control parameters	The control reception frame acknowledges all the communication parameters that were received from the peer.

Field	Description
Piafs layer up & Main FSM set to DATA	The PIAFS protocol is active on the router. The router is ready to receive data from the peer device.
Compression v42bis enabled	The compression protocol v42bis is enabled.
V42BIS:v42bis_init()	The v42bis compression protocol has been initiated.
V42BIS:Negotiated Values for P1, P2 are - 4096 , 250	In this example, P1 is the total count of encoded words when v42bis compression is enabled. P2 is the maximum letter line length for the V42bis compression.
Incoming call invoking ISDN_CALL_CONNECT	An incoming ISDN call connection message is received.
PPP	The PPP layer on the router becomes active and starts to process the PPP frame from the peer PIAFS device.

debug platform 6rd

To enable debugging for all IPv6 rapid deployment related occurrences on the Cisco 7600 router, and report on errors that occur for IPv6 rapid deployment, use the **debug platform 6rd** command in the privileged EXEC configuration mode. To disable the debugging, use the **no** form of the command .

debug platform 6rd {**events** | **errors**}

no debug platform 6rd {**events** | **errors**}

Syntax Description

events	Displays the debugging output for all IPv6 rapid deployment related occurrences on the router such as the creation of adjacencies, or the setting of the tunnel end-point.
errors	Displays the debugging output for problems related to IPv6 rapid deployment tunnel of RP.

Command Default

None

Command Modes

Privileged EXEC

Command History

Release	Modification
15.3.(2)S	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Use the debug command only to troubleshoot specific problems, or during troubleshooting sessions with Cisco technical support staff.

Example of Debugging Output for Events

The following shows sample output for events debugging:

```
CE1#debug platform 6rd events
6rd Events debugging is on
CE1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CE1(config)#int tunn56
CE1(config-if)#sh
CE1(config-if)#^Z
CE1#
*Mar  1 00:14:39.825 IST: cwan_release_6rd_tunnel_endpt: Released tunnel endpt 55
*Mar  1 00:14:41.041 IST: %SYS-5-CONFIG_I: Configured from console by console
CE1#
CE1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CE1(config)#int tunn56
CE1(config-if)#no sh
CE1(config-if)#^Z
CE1#
*Mar  1 00:14:59.013 IST: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:14:59.645 IST: cwan_6rd_tun_adj_attach_info: event for prefix [::]
*Mar  1 00:14:59.645 IST: cwan_adjacency_set_6rd_tunnel_endpoint: dest ip is
[0.0.0.0]
*Mar  1 00:14:59.645 IST: cwan_6rd_tun_adj_attach_info: final results:
Tunnel56 tunnel adj update, , ltl = 0x83  set, adj_handle [0x50201CD0]
*Mar  1 00:14:59.645 IST:  Tunnel end point ID[0] Active flag[1]
```

```
Source IP[100.0.56.1] Destination IP[0.0.0.0] Tunnel Vlan[1069]
Tunnel I/f no[102] Physical Vlan[1192]
Source MAC[0000.0000.0000] Dest MAC[0013.80b4.1c40]
*Mar  1 00:14:59.665 IST: cwan_6rd_tun_adj_attach_info: event for prefix
[2001:B000:6438::1]
*Mar  1 00:14:59.665 IST: cwan_adjacency_set_6rd_tunnel_endpoint: dest ip is
[100.100.56.1]
*Mar  1 00:14:59.665 IST: cwan_get_6rd_tunnel_endpt: Allocated tunne endpt 111
*Mar  1 00:14:59.665 IST: cwan_6rd_tun_adj_attach_info Cleared pending flag
                               tun_endpt->tunnel endpt 111
*Mar  1 00:14:59.665 IST: cwan_6rd_tun_adj_attach_info: final results: Tunnel56
tunnel adj update, GigabitEthernet3/4, ltl = 0x83  set, adj_handle [0x50201B10]
*Mar  1 00:14:59.665 IST:  Tunnel end point ID[111] Active flag[1]
Source IP[100.0.56.1] Destination IP[100.100.56.1] Tunnel Vlan[1069]
Tunnel I/f no[102] Physical Vlan[1310]
Source MAC[0013.80b4.1c40] Dest MAC[001c.b0ca.2240]
```

Example of Debugging Output for Errors

The following shows sample output for errors debugging:

```
CE1#debug platform 6rd errors
6rd Errors debugging is on
CE1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CE1(config)#int tunn56
CE1(config-if)#sh
CE1(config-if)#^Z
CE1#
*Mar  1 09:49:17.963 IST: cwan_release_6rd_tunnel_endpt: tunnel endpt 0 out of
range(1,8000)
*Mar  1 09:49:18.707 IST: %SYS-5-CONFIG_I: Configured from console by console
CE1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CE1(config)#int tunn56
CE1(config-if)#no sh
CE1(config-if)#^Z
CE1#
*Mar  1 09:49:45.603 IST: %SYS-5-CONFIG_I: Configured from console by console
```

debug platform condition

To filter debugging output for certain **debug** commands on the basis of specified conditions, use the **debug platform condition** command in privileged EXEC mode. To remove the specified condition, use the **no** form of this command.

debug platformcondition [interface *interface* **] { [mpls | access-list***access-list name* **] | [ipv4***ipv4-address/subnet-mask*| **ipv6***ipv6-address/subnet-mask* **][ingress | egress] }**
no debug platform condition

Syntax Description

interface *interface*	Filters output on the basis of the interface specified.
mpls	Enables conditional debug for MPLS packets.
access-list *access-list name*	Filters output on the basis of the specified access list.
ipv4 *ipv4-address/subnet-mask*	Filters output on the basis of the specified IPv4 address.
ipv6 *ipv6-address/subnet-mask*	Filters output on the basis of the specified IPv6 address.
ingress	Filters output on the basis of incoming packets.
egress	Filters output on the basis of outgoing packets.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 3.10	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Usage Guidelines

Use the **debug platform condition** command to generate output only for interfaces associated with a specified keyword.

The access list and IP address are mutually exclusive. If neither access list nor IP address is specified, all the packets are marked for debugging or packet trace.

The following example shows how to enable debug for packets that match access list 100 and destination IPv4 address 10.1.1.1 on the interface Gi0/0/1:

```
Router# access-list 100 permit ip any 10.1.1.1
Router# debug platform condition interface Gi0/0/1 access-list 100
```

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition feature	Enables conditional debugging for the specified feature.

Command	Description
debug platform condition start	Starts conditional debugging on a system.
debug platform condition stop	Stops conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to a platform.

debug platform condition

To filter debugging output for certain **debug** commands on the basis of specified conditions, use the **debug platform condition** command in privileged EXEC mode. To remove the specified condition, use the **no** form of this command.

debug platformcondition [interface *interface*] { [**mac**] [**access-list** *mac acl*] [**both** | **ingress** | **egress**] }
no debug platform condition

Syntax Description

interface *interface*	Filters output on the basis of the interface specified.
mac	Specifies the mac address.
access-list *access-list name*	Filters output on the basis of the specified access list.
both	(Optional) Filters output on the basis of the incoming and outgoing packets.
ingress	(Optional) Filters output on the basis of incoming packets.
egress	(Optional) Filters output on the basis of outgoing packets.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 16.11	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers, Cisco ISR 1000 Series Integrated Services Routers, Cisco ISR 4000 Series Integrated Services Routers, Cisco CSR 1000 Series Cloud Services Routers, and WCL.

Usage Guidelines

Use the **debug platform condition** command to generate output only for interfaces associated with a specified keyword.

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition feature	Enables conditional debugging for the specified feature.
debug platform condition start	Starts conditional debugging on a system.
debug platform condition stop	Stops conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to a platform.

debug platform condition match

To enable the conditional debugging filter to match the traffic flow in both directions, use the **bidirection** option with the **debug platform condition match** command in the privileged EXEC mode.

debug platform condition match { ipv4 | ipv6 | mac } source address destination address { both | ingress | egress } [bidirection]

Syntax Description

ipv4	(Optional) Uses IPv4 address as conditional filter.
ipv6	(Optional) Uses IPv6 address as conditional filter.
condition	Uses condition to enable platform conditional debugging
match	Uses match to set an inline conditional filter.
mac	Uses MAC address as conditional filter.
source	Sets the source address of the conditional filter.
destination	Sets the destination address of the conditional filter.
egress	Sets debugging in the egress direction.
ingress	Sets debugging in the ingress direction.
host	Uses the single host address as a conditional filter.
bidirection	Uses this option to enable bidirectional packet tracing.
both	Use this option to enable simultaneous ingress and egress conditional debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Cupertino 17.8.1a	The **bidirection** keyword was introduced to this command.

Usage Guidelines

The state of the number of packets that are being traced must be Healthy **H**. Use the **bidirection** option in the **debug platform condition** command to enable the conditional debugging filter to match the traffic flow in both directions.

Use the **show platform resources** command to verify used memory, which is total memory minus the accurate free memory.

Note We recommend that you check if the state is Healthy **H**. If the state appears to be "Critical (**C**)", you must be cautious about the number of packets that are being traced.

This example shows how to set the platform conditional debugging filter to match the bidirectional traffic for a given flow when using the packet-trace feature:

```
Router# debug platform condition match ipv4 host 192.0.2.6 host 198.51.100.1 both bidirection
```

This example displays all the debug settings:

```
Router#show debug

IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop

Conditions
     Direction
----------------------------------------------------------------------------------|---------
All Interfaces                          & IPV4 Filter [ALL PROTO] [host 192.0.2.6] [host
198.51.100.1] ] both bi


Feature Condition       Type                     Value
--------------------|---------------------|-------------------------------------------------


Feature      Type           Submode
                              Level
————————|—————————|————————————————————————————————————————————————————————————————|———————


IOSXE Packet Tracing Configs:
debug platform packet-trace packet 1024 circular fia-trace data-size 2048
debug platform packet-trace copy packet both size 2048 L2
```

Related Commands

Command	Description
show debugging	Displays both the platform conditions and the platform packet-trace configuration.
show platform packet-trace summary	Displays a summary of all the traced packets, with input and output interfaces, and processing result and reason.

debug platform condition feature

To enable conditional debugging for a specific feature, use the **debug platform condition feature** command in privileged EXEC mode. To disable the conditional debugging for a specific feature, use the **no** form of this command.

debug platform condition feature
feature-name [**controlplane** | **dataplane**] [*submode*] [**level** {**severe** | **warn** | **info** | **detail**}]
no debug platform condition feature

Syntax Description

feature-name	Name of the feature.
controlplane	Specifies control plane as the plane that the feature debug is applied on.
dataplane	Specifies data plane as the plane that the feature debug is applied on.
submode	Name of submode.
level	Specifies the level of the feature debug.
severe	Displays the severe debug messages.
warn	Displays the warning debug messages.
info	Displays information about the debug messages.
detail	Displays the detailed debug messages.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 3.10.0S	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Usage Guidelines

If the level is not specified, the level defaults to info. The severe debug level allows a feature to debug the events that led up to the severe event.

The following example shows how to enable conditional debug for the EVC feature. It also shows how to enable debug for packets that match access list 700 and specified MAC address on the interface Gi0/0/1 efp-id 100:

```
Router# access-list 700 permit 0000.0001.0002 0000.0000.0000
```

```
Router# debug platform condition interface Gi0/0/1 efp-id 100 access-list 700
Router# debug platform feature evc dataplane
Router# debug platform condition start
```

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition	Filters debugging output for **debug** commands on the basis of specified conditions.
debug platform condition start	Starts conditional debugging on a system.
debug platform condition stop	Stops conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to a platform.

debug platform condition feature alg dataplane submode

To enable conditional debugging, where debug messages related to specified connections are printed to the console, use the **debug platform condition feature alg dataplane submode** command in privileged EXEC mode. To disable conditional debugging, use the **no** form of this command.

debug platform condition feature alg dataplane submode [{**all** [**level** {**error** | **info** | **verbose** | **warning**}] | *protocol-name* [**. . .**[*protocol-name*]]}]

no debug platform condition feature alg dataplane submode [{**all** [**level** {**error** | **info** | **verbose** | **warning**}] | *protocol-name* [**. . .**[*protocol-name*]]}]

Syntax Description

all	Specifies all supported protocols.
level	Displays debug log severity levels.
error	Displays error and firewall packet drop conditions.
info	Displays information about an event.
verbose	Displays all debug log messages.
warning	Displays warning debug messages.

protocol-name (Optional) Protocol name. Use one of the following values for the protocol argument:

- **dns**—Displays debug Domain Name System (DNS) ALG information in the QFP datapath.
- **ftp**—Displays debug FTP ALG information in the QFP datapath.
- **gtp**—Displays debug General Packet Radio Service (GPRS) Tunneling Protocol (GTP) AIC information in the QFP datapath.
- **h323**—Displays debug H.323 ALG information in the QFP datapath.
- **http**—Displays debug HTTP ALG information in the QFP datapath.
- **imap**—Displays debug Internet Message Access Protocol (IMAP) ALG information in the QFP datapath.
- **ldap**—Displays debug Lightweight Directory Access Protocol (LDAP) ALG information in the QFP datapath.
- **level**—Displays debug level information.
- **msrpc**—Displays debug Microsoft Remote Procedure Call (MSRPC) ALG information in the QFP datapath.
- **netbios**—Displays debug Network Basic Input Output System (NetBIOS) ALG information in the QFP datapath.
- **pop3**—Displays debug Post Office Protocol 3 (POP3) AIC information in the QFP datapath.
- **pptp**—Displays debug Point-to-Point Tunneling Protocol (PPTP) ALG information in the QFP datapath.
- **rcmd**—Displays debug RCMD ALG information in the QFP datapath.
- **rtsp**—Displays debug Rapid Spanning Tree Protocol (RSTP) ALG information in the QFP datapath.
- **sip**—Displays debug Session Initiation Protocol (SIP) ALG information in the QFP datapath.
- **skinny**—Displays debug Skinny Client Control Protocol (SCCP) ALG information in the QFP datapath.
- **smtp**—Displays debug Simple Mail Transfer Protocol (SMTP) AIC information in the QFP datapath.
- **sunrpc**—Displays debug Sun RPC ALG-AIC information in the QFP datapath.
- **tftp**—Displays debug TFTP ALG information in the QFP datapath.
- **vtcp**—Displays debug VTCP information in the QFP datapath.

Command Default Info level is the default severity level that is logged.

Command Modes Privileged EXEC mode

Command History

Release	Modification
Cisco IOS XE Release 3.13S	This command was introduced.

Usage Guidelines

The application-layer gateway (ALG) type must be specified.

The following example shows how to enable conditional debugging for FTP:

```
Device# debug platform condition feature alg dataplane submode ftp
```

The following example shows how to enable conditional debugging for all supported protocols:

```
Device# debug platform condition feature alg dataplane submode all
```

debug platform condition feature fw controlplane level

To enable control plane conditional debugging for zone-based firewall, use the **debug platform condition feature fw controlplane level**

debug platform condition feature fw controlplane level {**error** | **info** | **verbose** | **warning**}[{**level**}]

Syntax Description

error	Enables error debugging.
info	Enables information debugging.
verbose	Enables verbose debugging.
warning	Enables warning debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.14S	This command was introduced.

Usage Guidelines

Configure the**debug platform condition feature fw controlplane level** commands to display significant control plane errors, like failure to program the policy on the QFP due to resource failure and so on.

Example

Use the **debug platform condition feature fw controlplane level** to enable control plane conditional debugging.

```
Device(config)# zone-pair security hi2int source inside destination $ecurity hi2int source
 inside destination outside
Device(config-sec-zone-pair)# service-policy type inspect p1
Device(config-sec-zone-pair)# end

Device(#) debug platform condition feature fw controlplane level

The following is the output for the debug command as seen in the logs:

Debug Log: /tmp/fp/trace/cpp_cp_F0-0.log
05/23 10:32:10.110 [buginf]: (debug): [cpp-fw]: (info): API [cpp_fw_handle_zonepair_create]:
 zp hi2int(5) src 1 dst 2
05/23 10:32:10.110 [buginf]: (debug): [cpp-fw]: (info): insert zonepair table for src zone
 1 dst zone 2key 0x10002 idx 0xbd addr: 0x8967f020
05/23 10:32:34.048 [buginf]: (debug): [cpp-fw]: (info): cpp_fw_handle_cgm_bind: client=4,
op=0, num_levels=1, tid=2882382797

05/23 10:32:34.048 [buginf]: (debug): [cpp-fw]: (info): API: zp 5, cg 29456, class 13586849,
 action 0, tid 0xabcdabcd, fot 0x1000001.
05/23 10:32:34.048 [buginf]: (debug): [cpp-fw]: (info): add class name inside2outside for
cg 29456 class 13586849 in datapath. hash_idx 0x38 entry addr: 0x89fc7000
05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): cpp_fw_handle_cgm_bind: client=4,
op=0, num_levels=1, tid=2882382797

05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): API: zp 5, cg 29456, class 13586849,
```

```
 action 65535, tid 0xabcdabcd, fot 0x4000000.
05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): add action 65535 to the list for
txn class 13586849
05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): cpp_fw_handle_cgm_bind: client=4,
op=0, num_levels=1, tid=2882382797

05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): API: zp 5, cg 29456, class 1593,
action 65535, tid 0xabcdabcd, fot 0x4000000.
05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): add class name class-default for
cg 29456 class 1593 in datapath. hash_idx 0x5f entry addr: 0x89fc7060
05/23 10:32:34.050 [buginf]: (debug): [cpp-fw]: (info): add action 65535 to the list for
txn class 1593
05/23 10:32:34.051 [buginf]: (debug): [cpp-fw]: (info): API [cpp_fw_handle_txn_commit]:
tid: 0xabcdabcd
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): op for class 13586849 cg 29456 in
txn 0xabcdabcd is 2(add). class_in_cg_pre_txn 0, with action 0. In this txn, num_txn_action
 1, null_action_bind 1, new_action_id 65535 class_modified 0
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): op for class 1593 cg 29456 in txn
0xabcdabcd is 2(add). class_in_cg_pre_txn 0, with action 0. In this txn, num_txn_action 1,
 null_action_bind 0, new_action_id 65535 class_modified 0
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): op for txn cg 29456 in txn 0xabcdabcd
 is 2. Before txn, num_class_in_cg 0. In txn, num_txn_class: 2, add 2 delete 0 class mod 0
 action change 0
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): op for txn zp 5 in txn 0xabcdabcd
is 2. num of cg attach/edit/detach: 1/0/0
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): attach cg 29456 zonepair 5, tid:
0xabcdabcd
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): cg_id is 29456, object_type: 0,
obj_id.ids[0]: 29456
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): Policy-map name: p1
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): Creating cg with name p1
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): analyze class 13586849 in cg 29456:
 num_proto 3 num_alg 0 algs[0] 0, has_alg 0
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): analyze class 1593 in cg 29456:
num_proto 0 num_alg 0 algs[0] 0, has_alg 1
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): [cpp_fw_hw_class_alloc]
class/action/cg_id/zp_id: 0x1149ab70, 0x10b1d7a0, 29456, 5
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 13586849 proto 0,
alloc stats blk 0x8fd45800
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 13586849 proto 1,
alloc stats blk 0x8fd45840
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 13586849 proto 2,
alloc stats blk 0x8fd45880
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 13586849 proto 3,
alloc stats blk 0x8fd458c0
05/23 10:32:34.052 [buginf]: (debug): [cpp-fw]: (info): DP rsrc for zp/class 5/13586849,
action: 0x1, filler/action/stats_tbl/tcp stats_blk ppe addr:
0x898b3400/0x8fd3e000/0x898b4000/0x8fd45840

05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): cpp_fw_hw_class_fill_l7_config:
l7_config = 0x0

05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): [cpp_fw_hw_class_alloc]
class/action/cg_id/zp_id: 0x1149af08, 0x10b1d7a0, 29456, 5
05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 1593 proto 0, alloc
 stats blk 0x8fd45900
05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 1593 proto 1, alloc
 stats blk 0x8fd45940
05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 1593 proto 2, alloc
 stats blk 0x8fd45980
05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): for zp 5 class 1593 proto 3, alloc
 stats blk 0x8fd459c0
05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): DP rsrc for zp/class 5/1593, action:
 0x81, filler/action/stats_tbl/tcp stats_blk ppe addr:
```

```
0x898b3418/0x8fd3e0f0/0x898b4120/0x8fd45940

05/23 10:32:34.053 [buginf]: (debug): [cpp-fw]: (info): cpp_fw_hw_class_fill_l7_config:
l7_config = 0x0

05/23 10:32:34.056 [buginf]: (debug): [cpp-fw]: (info): received fm op cb. status: 0, task_h:
 620607, ctx: 0x11496218 zp_id: 5, cg_id: 29456, op: 2
05/23 10:32:34.056 [buginf]: (debug): [cpp-fw]: (info): in txn 0xabcdabcd, async req: 1,
async reply so far: 1
05/23 10:32:34.056 [buginf]: (debug): [cpp-fw]: (info): On zonepair 5, cg 29456 has no alg
 enabled
05/23 10:32:34.056 [buginf]: (debug): [cpp-fw]: (info): update zonepair table entry. src 1
 dst 2 key/idx: 0x10002/0xbd with cce_info 00010008 00084441
05/23 10:32:34.056 [buginf]: (debug): [cpp-fw]: (info): post processing for txn 0xabcdabcd
05/23 10:32:34.057 [buginf]: (debug): [cpp-fw]: (info): cpp_fw_txn_post_process_l7 completes.
05/23 10:32:34.057 [buginf]: (debug): [cpp-fw]: (info): post process cg 29456 for zp 5 in
txn 0xabcdabcd
05/23 10:32:34.058 [buginf]: (debug): [cpp-fw]: (info): After txn 0xabcdabcd, cg 29456 has
 2 class in it.

Debug Log: /tmp/fp/trace/ fman-fp_F0-0.log
05/23 10:32:10.109 [fw]: (info): Added zone_pair hi2int (index 5, src inside (1), dest
outside (2))
05/23 10:32:10.109 [buginf]: (debug): [cpp-fw]: (info): FW API: cpp_fw_zonepair_create_a
reply: 0(Success)
05/23 10:32:10.109 [fw]: (info): Request CPP to create zone_pair 5 - Success
05/23 10:32:10.110 [buginf]: (debug): [cpp-fw]: (info): [cpp_fw_async_rsp_handler]: got
msg_type/rc/context 107/0/0xda395
05/23 10:32:10.110 [fw]: (info): CPP create for zone_pair hi2int (idx 5) - Success
05/23 10:32:34.046 [fw]: (info): Action and AOM objs filled for action ("(null)", 0) to
zonepair "hi2int"
05/23 10:32:34.046 [fw]: (info): Action and AOM objs filled for action ("(null)", 65535)
to zonepair "hi2int"
05/23 10:32:34.046 [fw]: (info): Action and AOM objs filled for action ("(null)", 65535)
to zonepair "hi2int"
05/23 10:32:34.047 [buginf]: (debug): [cpp-fw]: (info): Notification from CGM to FW, client:
 4 op: 13, batch_id: 2882382797, async: 1, ctx: 0x24
05/23 10:32:34.047 [buginf]: (debug): [cpp-fw]: (info): FW API: cpp_fw_cgm_bind_a reply:
0(Success)
05/23 10:32:34.048 [buginf]: (debug): [cpp-fw]: (info): Notification from CGM to FW, client:
 4 op: 13, batch_id: 2882382797, async: 1, ctx: 0x25
05/23 10:32:34.048 [buginf]: (debug): [cpp-fw]: (info): FW API: cpp_fw_cgm_bind_a reply:
0(Success)
05/23 10:32:34.049 [buginf]: (debug): [cpp-fw]: (info): [cpp_fw_async_rsp_handler]: got
msg_type/rc/context 117/0/0x24
05/23 10:32:34.049 [buginf]: (debug): [cpp-fw]: (info): Notification from CGM to FW, client:
 4 op: 13, batch_id: 2882382797, async: 1, ctx: 0x26
05/23 10:32:34.049 [buginf]: (debug): [cpp-fw]: (info): FW API: cpp_fw_cgm_bind_a reply:
0(Success)
```

debug platform condition feature multicast controlplane level

To enable control plane conditional debugging for multicast, use the **debug platform condition feature multicast controlplane level**

debug platform condition feature multicast controlplane level { **error** | **info** | **verbose** | **warning** }

Syntax Description

error	Enables error debugging.
info	Enables information debugging.
verbose	Enables verbose debugging.
warning	Enables warning debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 17.3.1	This command was introduced.

Usage Guidelines

Use the **debug platform condition feature multicast controlplane level** command to conifgure the debug commands for multicast via CONFD or NetConf.

Use the **debug platform condition feature multicast controlplane level** to enable control plane conditional debugging.

```
Router#debug platform condition feature multicast controlplane level error
Router#
```

Enable this debug command via CONFD

```
Router# debug platform { condition { feature { multicast { controlplane { level { error }
} } } } } }
result RPC request successful
Router#
```

debug platform condition feature multicast dataplane

Use the **debug platform condition feature multicast dataplane** command to enable conditional debugging in the dataplane IPv4/IPv6 multicast feature. To disable conditional debugging, use the **no** form of this command.

debug platform condition feature multicast dataplane v4mcast submode all boundary config error packet sr state

debug platform condition feature multicast dataplane v6mcast submode all boundary config error packet sr state

Syntax Description

all	Specifies all the information.
boundary	Displays the boundary information.
config	Displays the eonfiguration information for this feature.
error	Displays the error logs.
sr	Displays the sr messages.
state	Displays the state.

Command Modes

Privileged EXEC mode

Command History

Release	Modification
Cisco IOS XE Release 17.3.1	This command was introduced.

Example

```
Router#$debug platform condition feature multicast dataplane v4mcast submode error
Router#
```

Use this command to enable this command via CONFD:

```
Router# debug platform { condition { feature { multicast { dataplane { v4mcast { submode {
 error } } } } } } }
result RPC request successful
Router#
```

debug platform condition match

To filter MAC debugging output for certain **debug** commands on the basis of specified conditions, use the **debug platform condition match** command in privileged EXEC mode. To remove the specified condition, use the **no** form of this command.

debug platform condition [**interface** *interface*] **match** [{**mac** | **src mac** | **src mac mask** | **destination mac** | **destination mac mask** | **ethertype** *0-65535*}] [{**both** | **ingress** | **egress**}]

Syntax Description

interface *interface*	Filters output on the basis of the interface specified.
match	Enables conditional debug for matching packets.
src mac	(Optional) Specifies the src mac address.
src mac mask	(Optional) Specifies the src mac subnet mask.
destination mac	(Optional) Specifies the destination mac address.
destination mac mask	(Optional) Specifies the destination mac subnet mask.
ethertype	(Optional) Specifies the ether type.
both	(Optional) Filters output on the basis of both incoming and outgoing packets.
ingress	(Optional) Filters output on the basis of incoming packets.
egress	(Optional) Filters output on the basis of outgoing packets.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 16.11	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers, Cisco ISR 4000 Series Integrated Services Routers, Cisco ISR 1000 Series Integrated Services Routers, Cisco CSR 1000 Series Cloud Services Routers, and WCL.

Usage Guidelines

Use the **debug platform condition match** command to filter mac.

Conditional debug can match packets with the source MAC, source MAC mask, destination MAC and destination MAC mask.

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition feature	Enables conditional debugging for the specified feature.
debug platform condition start	Starts conditional debugging on a system.

Command	Description
debug platform condition stop	Stops conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to a platform.

debug platform condition match protocol

To filter IPv4 and IPv6 debugging output for certain **debug** commands on the basis of specified conditions, use the **debug platform condition match protocol** command in privileged EXEC mode. To remove the specified condition, use the **no** form of this command.

debug platform condition interface *interface name* **match** [{ **ipv4 | ipv6** }] **protocol** [{ **tcp** | **udp** | **protocol_id** }] [{ **src ip** | **src ip mask** | **src port** | **destination ip** | **destination ip mask** | **destination port** }] [{ **both** | **ingress** | **egress** }]
no debug platform condition match protocol

Syntax Description

interface *interface*	Filters output on the basis of the interface specified.
match	Enables conditional debug for matching packets.
IPv4	(Optional) Filters output on the basis of the specified IPv4 address.
Ipv6	(Optional) Filters output on the basis of the specified IPv6 address.
protocol	Filters output on the basis of the specified protocol.
tcp	(Optional) Specify tcp to filters output on the basis of the tcp.
udp	(Optional) Specifies udp to filters output on the basis of the udp.
protocol_id	(Optional) Specifies protocol id to filters output on the basis of the protocol id.
src ip	(Optional) Specifies the src ip address to filter output on the basis of the src ip.
src ip mask	(Optional) Specifies the src ip subnet mask to filter output on the basis of the src ip subnet mask.
destination ip	(Optional) Specifies the destination ip address to filter output on the basis of the destination ip address.
destination ip mask	(Optional) Specifies the destination ip address to filter output on the basis of the destination ip subnet mask.
destination port	(Optional) Specifies the destination port address to filter output on the basis of the destination port.
both	(Optional) Filters output on the basis of both incoming and outgoing packets.
ingress	(Optional) Filters output on the basis of incoming packets.
egress	(Optional) Filters output on the basis of outgoing packets.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 16.11	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers, Cisco ISR 1000 Series Integrated Services Routers, Cisco ISR 4000 Series Integrated Services Routers, Cisco CSR 1000 Series Cloud Services Routers, and WCL.

Usage Guidelines

Use the **debug platform condition match protocol** command to generate output only for interfaces associated with a specified keyword.

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition feature	Enables conditional debugging for the specified feature.
debug platform condition start	Starts conditional debugging on a system.
debug platform condition stop	Stops conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to a platform.

debug platform condition start

To start conditional debugging on a system, use the **debug platform condition start** command in privileged EXEC mode.

debug platform condition start

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 3.10	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

Example 1

The following example shows how to start conditional debugging on a system:

```
Router# debug platform condition interface Gi0/0/1 efp-id 100 access-list 700
Router# debug platform feature evc dataplane
Router# debug platform condition start
```

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition feature	Enables conditional debugging for the specified feature.
debug platform condition	Filters debugging output for **debug** commands on the basis of specified conditions.
debug platform condition stop	Stops conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to the platform.

debug platform condition stop

To stop conditional debugging on a system, use the **debug platform condition stop** command in privileged EXEC mode.

debug platform condition stop

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE 3.10	This command was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.

The following example shows how to stop conditional debugging on a system.

```
Router# debug platform condition interface Gi0/0/1 efp-id 100 access-list 700
Router# debug platform feature evc dataplane
Router# debug platform condition start
Router# debug platform condition stop
```

Related Commands

Command	Description
show platform condition	Displays the currently active debug configuration.
debug platform condition feature	Enables conditional debugging for the feature you specify.
debug platform condition	Filters debugging output for **debug** commands on the basis of specified conditions.
debug platform condition start	Starts conditional debugging on a system.
clear debug platform condition all	Removes the debug conditions applied to the platform.

debug platform hardware qfp active feature evtmon

To debug the event monitoring features in the Cisco QuantumFlow Processor (QFP), use the **debug platform hardware qfp feature evtmon** command in Privileged EXEC mode. To disable this form of debugging, use the **no** form of this command.

debug platform hardware qfp {**active** | **standby**} **feature evtmon** {**client debug-level** | **datapath protocol**}
no debug platform hardware qfp {**active** | **standby**} **feature evtmon** {**client debug-level** | **datapath protocol**}

Syntax Description

active	**Enables debug logging for the active processor.**
standby	**Enables debug logging for the standby processor.**
evtmon	Displays the event monitoring information pertaining to the processor.
client	Specifies the event monitoring QFP client information for one of the following debug-level options: • all • error • info • trace • warning
datapath	*Specifies the event monitoring datapath for one of the following protocols:* • *ip* --ipv4 protocol • *ipv6* --ipv6 protocol

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced on the Cisco ASR 1000 Series Routers.

Examples

The following example shows how to debug the event monitoring datapath for an IPv4 protocol: :

```
Router# debug platform hardware qfp active feature evtmon datapath ip
The selected EVTMON Datapath debugging is on
```

debug platform hardware qfp active feature ipsec

To display debugging information for IPsec events and counters in the Cisco Quantum Flow Processor (QFP) client, use the **debug platform hardware qfp active feature ipsec** command in privileged EXEC mode. To disable the display of this debugging information, use the **no** form of this command.

debug platform hardware qfp active feature ipsecdebug platform hardware qfp active feature ipsec {**client** {**error** | **info** | **trace** | **warning**} | **counter read-only** | **datapath** {**cce** | **droptype** *drop-type-number* | **error** | **info** | **pktcorrupt** *maximum-number* | **trace** | **warning**}}
no debug platform hardware qfp active feature ipsec {**client** {**error** | **info** | **trace** | **warning**} | **counter read-only** | **datapath** {**cce** | **droptype** *drop-type-number* | **error** | **info** | **pktcorrupt** *maximum-number* | **trace** | **warning**}}

Syntax Description

client	Enables debugging of IPsec events in the QFP client.
error	Enables debugging of errors.
info	Enables debugging of information.
trace	Enables debugging of packet tracing.
warning	Enables debugging of warnings.
counter	Enables debugging of IPsec counter settings in the QFP client.
read-only	Sets the debugging level of IPsec counter settings to read-only.
datapath	Enables debugging of IPsec events in the QFP datapath.
cce	Enables debugging of the IPsec common classification engine (CCE) in IPsec events.
droptype *drop-type-number*	Enables debugging of packet drop types in IPsec events. The range is from 1 to 69.
pktcorrupt *maximum-number*	Enables debugging of corrupt packets in QFP datapath IPsec events. The range for the maximum number of corrupt packets that are dumped is from 1 to 255.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.7S	This command was introduced.

Usage Guidelines

If you enter the **no debug all** command, debugging of the IPsec platforms is disabled.

Examples

The following example shows how to enable debugging for the IPsec datapath in QFP:

```
Device# debug platform hardware qfp active feature ipsec datapath cce

CPP IPSEC DATAPATH debugging is on
```

debug platform hardware qfp active feature wccp

To enable debug logging for the Web Cache Communication Protocol (WCCP) client in the Cisco Quantum Flow Processor (QFP), use the **debug platform hardware qfp active feature wccp** command in privileged EXEC mode. To disable WCCP QFP debug logging, use the **no** form of this command.

debug platform hardware qfp active feature wccp {{**client** | **lib-client** {**all** | **error** | **info** | **trace** | **warning**}} | **datapath all**}
no debug platform hardware qfp active feature wccp {{**client** | **lib-client** {**all** | **error** | **info** | **trace** | **warning**}} | **datapath all**}

Syntax Description

client	Enables WCCP QFP client debug logging.
lib-client	Enables WCCP QFP client-library debug logging.
all	Enables all logs.
error	Enables error logs.
info	Enables info logs.
trace	Enables trace logs.
warning	Enables warning logs.
datapath all	Enables all WCCP QFP datapath debug logging.

Command Default

WCCP QFP debug logging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Usage Guidelines

When the **debug platform hardware qfp active feature wccp** command is configured, QFP client debugs are enabled and can be collected from the forwarding processor (FP) from the file cpp_cp_F0-0.log.

When the **debug platform hardware qfp active feature wccp lib-client all** command is configured,QFP lib-client debugs are enabled and can be collected from the FP from the file fman-fp_F0-0.log.

When the **debug platform hardware qfp active feature wccp datapath all**command is configured, QFP datapath debugs are enabled and can be collected from the FP from the file cpp_cp-F0-0.log.

Examples

The following is sample output from the **debug platform hardware qfp active feature wccp** command:

```
Router# debug platform hardware qfp active feature wccp
```

A WCCP service is configured:

```
06/17 10:48:15.980 [(null)]: (debug): cpp_wccp_service_add_handler: service_params::: type
 =0 id = 0priority = 240  is_closed = 0 assign = 0
06/17 10:48:15.980 [(null)]: (debug): cpp_wccp_dplane_init dplane cpp-init for all cpps
06/17 10:48:15.980 [(null)]: (debug): cpp_wccp_dplane_init_cpp Enter: cpp_info = 0x1027b970:
.
.
.
```

The sequence of messages repeats for each access control entry (ACE) of a merged access control list (ACL):

```
06/17 10:53:38.792 [(null)]: (debug): cpp_wccp_update_bind_obj_list:idx = 63 bind-info:no.lvl
 = 1 fobj = 80024000 bind-id = 0
06/17 10:53:38.792 [(null)]: (debug): cpp_wccp_update_bind_obj_list fobj:service-id = 0
type = 0 cache-id = 9action = 2 acl-log = 0
06/17 10:53:38.792 [(null)]: (debug): cpp_wccp_add_dplane_cache_desc service-index = 0,
cache_id = 9
06/17 10:53:38.792 [(null)]: (debug): cpp_wccp_get_dplane_cache_index service-index = 0,
cache_id = 9
06/17 10:53:38.792 [(null)]: (debug): cpp_wccp_create_dplane_cache_index Cache index = 0
exists for cache-id = 9,service-index  = 0
.
.
.
```

WCCP redirection is configured on an interface:

```
06/17 13:15:44.655 [(null)]: (debug): cpp_wccp_intf_attach_msg req = 0x13116848, msg-len =
 36
06/17 13:15:44.655 [(null)]: (debug): cpp_wccp_intf_attach_handler: type = 0 id = 0 ifh =
17dir = 0 vrfid = 0
06/17 13:15:44.655 [(null)]: (debug): cpp_wccp_get_service_index WCCP: service_id 0 vrfid
0service_desc_index 0
06/17 13:15:44.655 [(null)]: (debug): cpp_wccp_get_service_desc: service-id: 0  type = 0
index = 0
.
.
.
```

Debug messages appear for each ACE of the merged ACL for a service group:

```
06/17 13:15:44.670 [(null)]: (debug): cpp_wccp_translate_fobj_to_cce_result Entry
06/17 13:15:44.670 [(null)]: (debug): cpp_wccp_get_service_index WCCP: service_id 0 vrfid
0service_desc_index 0
06/17 13:15:44.670 [(null)]: (debug): cpp_wccp_get_service_desc: service-id: 0  type = 0
index = 0
06/17 13:15:44.670 [(null)]: (debug): cpp_wccp_get_dplane_cache_index service-index = 0,
cache_id = 9
.
.
.
```

Redirection is removed from an interface:

```
06/17 13:24:54.617 [(null)]: (debug): cpp_wccp_intf_detach_handler: type = 0 id = 0 ifh =
17dir = 0 vrfid = 0
06/17 13:24:54.617 [(null)]: (debug): cpp_wccp_get_service_index WCCP: service_id 0 vrfid
0service_desc_index 0
06/17 13:24:54.617 [(null)]: (debug): cpp_wccp_get_service_desc: service-id: 0  type = 0
index = 0
06/17 13:24:54.617 [(null)]: (debug): cpp_wccp_intf_detach_handler:hw_cg_node, ifh = 17 dir
```

```
 = 0vrfid = 0 service-index = 0 exists
.
.
```

A service group is unconfigured:

```
06/17 13:29:41.828 [(null)]: (debug): cpp_wccp_cache_delete_handler: cache-desc ip-addr =
5a140102 id-addr = 0cache-id = 9  cef_handle = 0x112d3b68 cef-obj-type = 10router-id =
42424242 ce_mac_addr  fwd-method = 0 hw-addr = 0x11188f78
06/17 13:29:41.828 [(null)]: (debug): cpp_wccp_remove_dplane_ip_hash_entry cache_id= 9:
06/17 13:29:41.828 [(null)]: (debug): cpp_wccp_remove_dplane_ip_hash_entry ip-hash-index =
 6934:
.
.
.
```

The following is sample output from the **debug platform hardware qfp active feature wccp lib-client all**command:

```
Router#
debug platform hardware qfp active feature wccp lib-client all
```

A WCCP service group is configured:

```
06/17 13:47:00.158 [buginf]: (debug): cpp_wccp_service_group_add_a: API call from PAL
service-type = 0 id = 0vrfid = 0, priority = 240 is_closed = 0 has_ports = 1 assign-method
 = 0
06/17 13:47:00.158 [buginf]: (debug): cpp_wccp_api_async_msg_send: data size = 28 for this
 3message
06/17 13:47:00.158 [buginf]: (debug): cpp_wccp_api_async_send_cb: SMC async send call-back
.
.
.
```

The set of debug messages repeats for each ACE of the merged ACL of the WCCP service group:

```
06/17 13:47:29.474 [buginf]: (debug):   Notification from CGM to WCCP, op:13, tid:0,async:
 0, ctx: (nil)
06/17 13:47:29.474 [buginf]: (debug): cpp_wccp_cgm_notif_handler:cgm BIND num_lvl = 1,
bind-id = 0 fobj = 80028000
06/17 13:47:29.474 [buginf]: (debug):   Notification from CGM to WCCP, op:2, tid:0,async:
1,ctx: 0x77
.
.
.
```

WCCP redirection is configured on an interface:

```
06/17 13:52:05.841 [buginf]: (debug):   Notification from CGM to WCCP, op:1, tid:0,async:
0,ctx: (nil)
06/17 13:52:05.841 [buginf]: (debug): cpp_wccp_attach_service_to_intf_a: API call from PAL
 service-type = 0 id = 0 vrfid = 0 if_h = 11 dir = 0
06/17 13:52:05.841 [buginf]: (debug): cpp_wccp_attach_service_to_intf_a:tid el= 0x11347470
 ifh = 17, dir = 0 id = 0 type = 0 vrfid = 0
.
.
.
```

WCCP is unconfigured on an interface:

```
06/17 13:54:30.544 [buginf]: (debug):   Notification from CGM to WCCP, op:1, tid:0,async:
```

```
0,ctx: (nil)
06/17 13:54:30.544 [buginf]: (debug): cpp_wccp_detach_service_from_intf_a: API call from
PALservice-type = 0 id = 0 vrfid = 0 if_h = 11 dir = 0
06/17 13:54:30.544 [buginf]: (debug): cpp_wccp_detach_service_from_intf_a:tid el=
0x11338890ifh = 17, dir = 0 id = 0 type = 0
06/17 13:54:30.544 [buginf]: (debug):   Notification from CGM to WCCP, op:2, tid:0,async:
1,ctx: 0x79
.
.
.
```

A WCCP service group is unconfigured:

```
06/17 13:56:14.492 [buginf]: (debug): cpp_wccp_cache_delete_a: API call from PAL cache-id=
 10
06/17 13:56:14.492 [buginf]: (debug): cpp_wccp_api_async_msg_send: data size = 2 for this
6 message
06/17 13:56:14.492 [buginf]: (debug): cpp_wccp_api_async_send_cb: SMC async send call-back
06/17 13:56:14.492 [buginf]: (debug):  cpp_wccp_api_async_msg_send successfully sent msg-type
 6 to server.
06/17 13:56:14.492 [buginf]: (debug):   Notification from CGM to WCCP, op:1, tid:0,async:
0,ctx: (nil)
06/17 13:56:14.492 [buginf]: (debug):   Notification from CGM to WCCP, op:14, tid:0,async:
 0, ctx: (nil)
06/17 13:56:14.493 [buginf]: (debug): cpp_wccp_cgm_notif_handler:cgm BIND num_lvl = 1,
bind-id = 0 fobj = 80028000
.
.
.
```

The debug messages repeat for each ACE of the merged ACL for the WCCP service group:

```
06/17 13:56:14.500 [buginf]: (debug):   Notification from CGM to WCCP, op:14, tid:0,async:
 0, ctx: (nil)
06/17 13:56:14.500 [buginf]: (debug): cpp_wccp_cgm_notif_handler:cgm BIND num_lvl = 1,
bind-id = 0 fobj = 80028000
06/17 13:56:14.501 [buginf]: (debug):   Notification from CGM to WCCP, op:2, tid:0,async:
1,ctx: 0x7a

.


.
.
```

The following is sample output from the **debug platform hardware qfp active feature wccp datapath all**command:

```
Router# debug platform hardware qfp active feature wccp datapath all
```

A packet is successfully redirected:

```
06/17 14:49:28.935 [(null)]: (debug):
QFP:00 Thread:090 TS:00003918904609765795
##########################################################################################
06/17 14:49:28.936 [(null)]: (debug):
QFP:00 Thread:090 TS:00003918904609777642 CCE IPV4 PKT
(src:3.3.3.2,dst:2.2.2.2,sprt:0000,dprt:0050,prot:06,tos:00,len:0014,ttl:3f) , intf:3f3
06/17 14:49:28.936 [(null)]: (debug):
QFP:00 Thread:090 TS:00003918904609814715
##########################################################################################
06/17 14:49:28.936 [(null)]: (debug):
```

```
QFP:00 Thread:090 TS:00003918904609825865 CCE IPV4 UIDB_INFO W0:00000004, W1:00084441,
tcam_region_index:0004, key_index:00, cmd:00084441
06/17 14:49:28.936 [(null)]: (debug):
.
.
.
```

Related Commands

Command	Description
clear ip wccp	Removes WCCP statistics (counts) maintained on the router for a particular service.
ip wccp	Enables support of the specified WCCP service for participation in a service group.
ip wccp check services all	Enables enable all WCCP services.
ip wccp outbound-acl-check	Enables execution of ACL applied on the actual outgoing interface of a packet before a decision is taken to redirect a packet.
ip wccp redirect	Enables packet redirection on an outbound or inbound interface using WCCP.

debug platform hardware qfp feature

To debug features in the Cisco QuantumFlow Processor (QFP), use the debug platform hardware qfp feature command in Privileged EXEC mode. To disable this form of debugging, use the **no** form of this command.

debug platform hardware qfp {**active** | **standby**} **feature alg** {**client debug-level** | **datapath protocol** [**detail**]}
no debug platform hardware qfp {**active** | **standby**} **feature alg** {**client debug-level** | **datapath netbios** [**detail**]}

Syntax Description

active	Enables debug logging for the active processor.
standby	Enables debug logging for the standby processor.
alg	Displays the Application Level Gateway (ALG) information of the processor.
client	Specifies the ALG QFP client information.
debug-level	One of the following debug level options: • **all** • **error** • **info** • **trace** • **warning** **Note** The debug level options are not supported in the following protocols: • dns • ftp • h323 • ldap • sip • skinny • rtsp • rcmd • tftp • netbios
datapath	Specifies the ALG datapath.

protocol	One of the following protocols: • **dns** • **ftp** • **h323** • **http** • **imap** • **ldap** • **netbios** • **pop3** • **rcmd** • **rtsp** • **sip** • **skinny** • **smtp** • **sunrpc** • **tftp**
detail	(Optional) Specifies the QFP datapath ALG in detail.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 2.2	This command was introduced.
Cisco IOS XE Release 3.1S	This command was modified. Support for the Network Basic Input Output System (NetBIOS) protocol. The following keywords were added: **netbios-dgm**,**netbios-ns**,**netbios-ssn**.
15.1(1)S	This command was integrated into Cisco IOS XE Release 15.1(1)S

Examples

The following example shows how to debug the ALG datapath for a dns protocol:

```
Router# debug platform hardware qfp  active feature alg datapath dns
CPP ALG datapath event debugging is on
```

Related Commands

Command	Description
show platform hardware qfp feature	Displays feature specific information in QFP.

debug platform hardware qfp feature otv client

To enable Overlay Transport Virtualization (OTV) debugging on the Quantum Flow Processor (QFP) client, use the **debug platform hardware qfp feature otv client** command in privileged EXEC mode. To disable logging of the debug messages, use the **no** form of this command.

debug platform hardware qfp {**active** | **standby**} **feature otv client** {**all** | **error** | **info** | **trace** | **warning**}
no debug platform hardware qfp {**active** | **standby**} **feature otv client** {**all** | **error** | **info** | **trace** | **warning**}

Syntax Description

active	Enables debug of the active instance of the processor.
standby	Enables debug of the standby instance of the processor.
all	Enables all debugging.
error	Enables error debugging.
info	Enables info debugging.
trace	Enables trace debugging.
warning	Enables warning debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following is sample output from the **debug platform hardware qfp feature otv client** command:

```
Router# debug platform hardware qfp feature otv client all

The output of the debug is saved on the tracelog file for cpp_cp_F0-0.log(or cpp_cp_F1-0.log):
[cpp_otv_ea_decap_unprovision:844] Entering
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_decap_unprovision:865] received decap unprovision message, is_async==1
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_decap_unprovision_cmn:434] cpp_ifhandle=741
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_decap_dp_unprovision:192] ifhandle=741 clear output subblock
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_decap_dp_unprovision:230] disable Overlay EFP feature cpp_ifhandle=7741
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_decap_unprovision_cmn:474] OTV decap chain unprovision success, cpp__ifhandle=741
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_msg_send_cb:47] Entering cpp_otv_ea_msg_send_cb
11/02 17:12:39.383 [(null)]: (debug):
[cpp_otv_ea_msg_send:104] send reply back to API LIB, async=1
11/02 17:12:39.384 [(null)]: (debug): m
[cpp_otv_ea_decap_unprovision:888] cpp_otv_ea_decap_unprovision retval=Success
```

Related Commands

Command	Description
show platform hardware qfp feature otv client interface	Displays OTV feature-specific information for the specified overlay interface

debug platform link-dc

To display debugging messages for the link daughter card, use the **debugplatformlink-dc**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform link-dc {**dwdm** | **interface** | **interrupt** | **netclk** | **serdes** | **transceiver** | **wanphy**}
no debug platform link-dc {**dwdm** | **interface** | **interrupt** | **netclk** | **serdes** | **transceiver** | **wanphy**}

Syntax Description

dwdm	OTN G.709/DWDM driver debug information.
interface	Interface driver debug information.
interrupt	Interrupt debug information.
netclk	Network clocking debug information.
serdes	Physical layer (PHY) and SerDes debug information.
transceiver	Pluggable optics module information.
wanphy	WAN PHY driver debug information.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRD	This command was introduced. **Note** This command applies only to the Cisco 7600 Series Ethernet Services Plus (ES+) line card on the Cisco 7600 series router.
12.2(33)SRD1	This command added the dwdm and wanphy keywords.

Usage Guidelines

Use this command with the remote command command or the attach command in privileged EXEC mode.

Examples

The following examples show the output for both the debug platform link-dc tranceiver command and the debug platform link-dc interrupt command. Notice that the show platform hardware transceiver command shows the status for the port.

```
Router# remote command module 1 debug platform link-dc tranceiver
Link-DC transceiver debugging is on
Router# remote command module 1 debug platform link-dc interrupt
Link-DC interrupt debugging is on
Router# remote command module 1 show debug
x40g subsystem:
  Link-DC transceiver debugging is on
  Link-DC interrupt debugging is on
Router# remote command module 1 show platform hardware transceiver status 1
Show status info for port 1:
```

```
TenGigabitEthernet1/1:
        State: Enabled
        Environmental Information - raw values
                Temperature: 7616
                Tx voltage: 0 in units of 100uVolt
                Tx bias: 28722 uA
                Tx power: -2 dBm (5441 in units of 0.1 uW)
                Rx power: 0 dBm (7712 in units of 0.1 uW)
                (AUX1) Laser Temperature: 8704
                (AUX2) +3.3V Supply Voltage: 32928
        XFP TX is enabled.
        XFP TX is soft enabled.
        XFP is ready.
        XFP is not power down.
        XFP is not soft power down.
        XFP doesn't have interrupt(s).
        XFP is not LOS.
        XFP data is ready.
        XFP TX path is ready.
        XFP TX laser is not in fault condition.
        XFP TX path CDR is locked.
        XFP RX path is ready.
        XFP RX path CDR is locked.
        No active alarms
        No active warning
Router-dfc1#
*Aug 15 11:20:26.436 PDT: DFC1: TenGigabitEthernet1/1 XFP: show status
*Aug 15 11:20:26.436 PDT: DFC1: TenGigabitEthernet1/1 XFP: show environmental monitoring
*Aug 15 11:20:26.436 PDT: DFC1: pluggable optics read - addr: 50, offset: 60, len: 14,
dataptr: 2377A668
*Aug 15 11:20:26.448 PDT: DFC1: pluggable optics read - addr: 50, offset: 6E, len: 2,
dataptr: 21AA028E
*Aug 15 11:20:26.452 PDT: DFC1: pluggable optics read - addr: 50, offset: 50, len: 2,
dataptr: 2377A6A0
*Aug 15 11:20:26.456 PDT: DFC1: pluggable optics read - addr: 50, offset: 52, len: 2,
dataptr: 2377A6A2
```

Note The following console log is seen when both the debug platform link-dc tranceiver command and the debug platform link-dc interrupt command are entered (as in the preceding example), and there is a transceiver Rx loss of signal (LOS) event.

```
Router-dfc1#
*Aug 15 11:23:52.127 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x8000
*Aug 15 11:23:52.127 PDT: DFC1: x40g_link_xphy_isr: xphy intr intr_st 0x80000
*Aug 15 11:23:52.127 PDT: DFC1: x40g_link_xphy_isr: xphy intr port 1
*Aug 15 11:23:52.127 PDT: DFC1: x40g_xphy_link_status_callout: port 1 link status 0
*Aug 15 11:23:52.131 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x8000
*Aug 15 11:23:52.131 PDT: DFC1: x40g_link_xphy_isr: xphy intr intr_st 0x80000
*Aug 15 11:23:52.131 PDT: DFC1: x40g_link_xphy_isr: xphy intr port 1
*Aug 15 11:23:52.131 PDT: DFC1: x40g_xphy_link_status_callout: port 1 link status 1
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_dc_process: interrupt msg_id 6, msg_num 1
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x8000
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_xphy_isr: xphy intr intr_st 0x80000
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_xphy_isr: xphy intr port 1
*Aug 15 11:23:52.135 PDT: DFC1: x40g_xphy_link_status_callout: port 1 link status 0
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x4000
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_xcvr_isr: intr_st 0x2, start 0, end 4, type
2,port_offset 0x0
*Aug 15 11:23:52.135 PDT: DFC1: Link xcvr port 1: Rx LOS interrupt
*Aug 15 11:23:52.135 PDT: DFC1: x40g_link_dc_process: interrupt msg_id 2, msg_num 1
```

```
*Aug 15 11:23:52.135 PDT: DFC1: Port 2: transceiver Rx LOS event
*Aug 15 11:23:52.147 PDT: DFC1: x40g_link_dc_process: xcvr oir timer timeout
00:12:37: %LINEPROTO-DFC1-5-UPDOWN: Line protocol on Interface TenGigabitEthernet1/2, changed
 state to down
*Aug 15 11:24:46.576 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x4000
*Aug 15 11:24:46.576 PDT: DFC1: x40g_link_xcvr_isr: intr_st 0x2, start 0, end 4, type
2,port_offset 0x0
*Aug 15 11:24:46.576 PDT: DFC1: Link xcvr port 1: Rx LOS interrupt
*Aug 15 11:24:46.576 PDT: DFC1: x40g_link_dc_process: interrupt msg_id 2, msg_num 1
*Aug 15 11:24:46.576 PDT: DFC1: Port 2: transceiver Rx LOS recovered
*Aug 15 11:24:46.580 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x8000
*Aug 15 11:24:46.580 PDT: DFC1: x40g_link_xphy_isr: xphy intr intr_st 0x80000
*Aug 15 11:24:46.580 PDT: DFC1: x40g_link_xphy_isr: xphy intr port 1
*Aug 15 11:24:46.580 PDT: DFC1: x40g_xphy_link_status_callout: port 1 link status 0
*Aug 15 11:24:46.584 PDT: DFC1: x40g_link_dc_interrupt_handler:  intr_status 0x8000
*Aug 15 11:24:46.584 PDT: DFC1: x40g_link_xphy_isr: xphy intr intr_st 0x80000
*Aug 15 11:24:46.584 PDT: DFC1: x40g_link_xphy_isr: xphy intr port 1
*Aug 15 11:24:46.584 PDT: DFC1: x40g_xphy_link_status_callout: port 1 link status 1
*Aug 15 11:24:46.584 PDT: DFC1: x40g_link_dc_process: interrupt msg_id 6, msg_num 1
*Aug 15 11:24:46.600 PDT: DFC1: x40g_link_dc_process: xcvr oir timer timeout
00:13:31: %LINEPROTO-DFC1-5-UPDOWN: Line protocol on Interface TenGigabitEthernet1/2, changed
 state to up
```

The following example shows the output for the debug platform link-dc dwdm command.

```
Router-dfc1# debug platform link-dc dwdm
Link-DC OTN G.709/DWDM debugging is on
*Jan 28 12:10:38.784 PDT: DFC1: Port 1: OTN Alarm Query, return ptr 228E877C
los 1, oof 0, lof 0, mfas 1, lom 0
otuAis 0, otuIae 0-0, otuBdi 0, otuTim 0
oduAis 0, oduBdi 0, oduLck 0, oduOci 0, oduPtim 0
*Jan 28 12:10:38.864 PDT: DFC1: x40g_link_pemaquid_pm_tick_timer_event(1): pm_tick timer
timeout
*Jan 28 12:10:39.364 PDT: DFC1: x40g_link_pemaquid_pm_tick_timer_event(1): pm_tick timer
timeout
*Jan 28 12:10:39.840 PDT: DFC1: Port 1: OTN Alarm Query, return ptr 228E877C
los 1, oof 0, lof 0, mfas 1, lom 0
otuAis 0, otuIae 0-0, otuBdi 0, otuTim 0
oduAis 0, oduBdi 0, oduLck 0, oduOci 0, oduPtim 0
```

The following example shows the output for the debug platform link-dc wanphy command.

```
Router-dfc1# debug platform link-dc wanphy
Link-DC WAN PHY debugging is on
*Jan 28 11:59:16.184 PDT: DFC1: Port 1 WIS alarms:
     ser 0, plm_p_far 0, ais_p_far 0, lof 0, los 0
     rdi 0, ais_l 0, lcd_p 0, plm_p 0, ais_p 0, lop 0

*Jan 28 11:59:17.184 PDT: DFC1: Port 1 WIS alarms:
     ser 0, plm_p_far 0, ais_p_far 0, lof 0, los 0
     rdi 0, ais_l 0, lcd_p 0, plm_p 0, ais_p 0, lop 0
*Jan 28 11:59:17.184 PDT: DFC1: Port 1 WIS counters: b1 0, b2 0, b3 0, fe_b2 0, fe_b3 0
*Jan 28 11:59:17.184 PDT: DFC1: Port 1 WIS J1RX: 0x0000000000000089.0x302E302E302E3000
...
*Jan 28 11:59:22.288 PDT: DFC1: Port 1 WIS alarms:
     ser 0, plm_p_far 0, ais_p_far 0, lof 0, los 0
     rdi 0, ais_l 0, lcd_p 0, plm_p 0, ais_p 0, lop 0
*Jan 28 11:59:22.288 PDT: DFC1: Port 1 WIS counters: b1 0, b2 0, b3 0, fe_b2 0, fe_b3 0
*Jan 28 11:59:22.288 PDT: DFC1: Port 1 WIS J1RX: 0x0000000000000089.0x302E302E302E3000
```

Related Commands

Command	Description
show platform hardware transceiver	Displays transceiver information on a port.

debug platform software evtmon

To debug the event monitoring features in the Cisco QuantumFlow Processor (QFP), use the **debug platform software evtmon** command in Privileged EXEC mode. To disable this form of debugging, use the **no** form of this command.

debug platform software evtmon configuration
no debug platform software evtmon configuration

Syntax Description

configuration	**Enables configuration-related debugs.**

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced on the Cisco ASR 1000 Series Routers.

Examples

The following example shows how to debug the event monitoring configurations:

```
Router# debug platform software evtmon configuration
evtmon configuration messages debugging is on
```

debug platform software l2fib

To enable Overlay Transport Virtualization (OTV) debugging on the Cisco IOS daemon (IOSd) for the Layer 2 Forwarding Information Base (L2FIB) object, use the **debug platform software l2fib** command in privileged EXEC mode. To disable logging of the debug messages, use the **no** form of this command.

debug platform software l2fib {**error** | **events** | **verbose**}
no debug platform software l2fib {**error** | **events** | **verbose**}

Syntax Description

error	Enables error debugging.
events	Enables event debugging.
verbose	Enables verbose debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following is sample output from the **debug platform software l2fib** command:

```
Router# debug platform software l2fib events

*Nov  2 16:41:37.593: FMANRP-L2fib: Print download message TDL:
message@l2fib_mlist_cfg: {
  l2fib_mlist@l2fib_mlist: {
    mlist_id@obj_id: {
      index@U32:4006
    }
    nlist@l2fib_nhop_list: {
      num_nhop@U32:1
      entry_cfg[0]@l2fib_nhop_update: {
        nhop@l2fib_nhop: {
          nhop_key@l2fib_nhop_key: {
            nhdl@U32:16033
          }
          nhop_type@l2fib_nhop_type:L2FIB_NHOP_TYPE_EFP
          nhop_type.efp@efp_idx: {
            bindidx@U32:0
          }
        }
        upd_type@l2fib_nhop_upd_type:L2FIB_NHOP_UPD_TYPE_DEL
      }
    }
  }
  cfg_action@cfg_action:MCP_CFG_ACTION_MODIFY
}
```

Related Commands

Command	Description
show platform software l2fib fp	Displays the global bridge domain table for MAC and Layer 2 multicast on the FMAN on the FP.
show platform software l2fib rp	Displays the global bridge domain table for MAC and multicast on the FMAN on the RP.

debug platform software multicast

To display information about log events, packet information, and assert events, use the **debug platform software multicast**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast {**all events** | **assert events**}
no debug platform software multicast {**all events** | **assert events**}

Syntax Description

all	Displays all multicast hardware switching debugging information, including errors, events, and packets for the specified group.
assert	Specifies the assert events.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from thedebug platform software multicastcommand using the all keyword:

```
PE-3-sp#debug platform software multicast all
Global enable but not the periodic debugging is on
PE-3-sp#
*Oct 30 09:17:26.150 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
*Oct 30 09:17:26.770 EDT: SP: hal_timer_event: NRPF-AG
*Oct 30 09:17:27.151 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
*Oct 30 09:17:28.151 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
*Oct 30 09:17:28.395 EDT: SP: hal_timer_event: NRPF-AG
*Oct 30 09:17:29.152 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
*Oct 30 09:17:30.152 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
*Oct 30 09:17:30.248 EDT: SP: hal_timer_event: NRPF-AGun al
*Oct 30 09:17:31.153 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
```

The following example shows output from the debug platform software multicastcommand using the assert keyword:

```
PE-3-sp#debug platform software multicast assert
Assertion for Layer 2 multicast debugging is on
PE-3-sp#
PE-3-sp#debug platform software multicast ha l2-sso all
Debug for mcast SSO all debugging is on
PE-3-sp#debug platform software multicast ha l2-sso err
PE-3-sp#debug platform software multicast ha l2-sso error
Debug for mcast SSO error debugging is on
PE-3-sp#debug platform software multicast ha l2-sso eve
```

```
PE-3-sp#debug platform software multicast ha l2-sso event
Debug for mcast SSO events debugging is on
PE-3-sp#debug platform software multicast ha l2-sso pak
PE-3-sp#debug platform software multicast ha l2-sso pak
Debug for mcast SSO packets debugging is on
PE-3-sp#
```

Related Commands

Command	Description
debug platform software multicast	Displays the multicast debugging information.

debug platform software multicast cgmp

To display information about cgmp debugging events and packet information use the **debug platform software multicast cgmp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast cgmp {**event events** | **pak events**}
debug platform software multicast cgmp {**event events** | **pak events**}

Syntax Description

event	Specifies the events for the selected group.
pak	Specifies the packet information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast cgmp** command using the event keyword:

```
PE-3-sp#debug platform software multicast cgmp event
Router Discovery (CGMP Protocol) event log debugging is on
```

The following example shows output from the **debug platform software multicast cgmp** command using the pak keyword:

```
PE-3-sp#debug platform software multicast cgmp pak
```

Router Discovery (CGMP Protocol) packet log debugging is on

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast igmp

To display information about igmp debugging events and packet information use the **debug platform software multicast igmp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast igmp {**event events** | **pak events**}
no debug platform software multicast igmp {**event events** | **pak events**}

Syntax Description

event	Specifies the igmp events for the selected group.
pak	Specifies the igmp packet information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast igmp** command using the event keyword:

```
Router# debug platform software multicast igmp event
PE-3-sp#debug platform software multicast igmp event
IGMP snooping event log debugging is on
...
```

The following example shows output from the **debug platform software multicast igmp** command using the pak keyword:

```
PE-3-sp#debug platform software multicast igmp pak
PE-3-sp#debug platform software multicast igmp pak
IGMP snooping packet log debugging is on
PE-3-sp#
*Oct 30 09:26:22.143 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035
*Oct 30 09:26:22.143 EDT: SP: Packet dump:
18000070:             0100 5E000016 00000E00        ..^.......
18000080: 02000800 45000028 00000000 400254BC  ....E..(....@.T<
18000090: 46000002 E0000016 2200CBF6 00000001  F...`...".Kv....
180000A0: 01000001 E8000104 28000002 00010203  ....h...(.......
```

180000B0: 04058C

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast ip cmfib

To display information about multicast ip cmfib errors, shortcut events, and export the hardware statistics command, use the **debug platform software multicast ip cmfib**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ip cmfib {**error** | **events** | **stats**}
no debug platform software multicast ip cmfib {**error** | **events** | **stats**}

Syntax Description

error	Specifies the mfib IPV4 error information.
event	Specifies the IPv4 shortcut event information.
stats	Specifies the IPV4 hardware statistic information for export.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast ip cmfib** command using the error keyword:

PE-3-sp#debug platform software multicast ip cmfib cmfib error

CMFIB-LC IPv6 error debugging enabled

The following example shows output from the **debug platform software multicast ip cmfib** command using the event keyword:

PE-3-sp#debug platform software multicast ip cmfib cmfib eve

CMFIB-LC IPv6 event debugging enabled

The following example shows output from the **debug platform software multicast ip cmfib** command using the stats keyword:

PE-3-sp#debug platform software multicast ip cmfib cmfib stats

CMFIB-LC IPv6 stats debugging enabled

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast ip cmfib error

To display information about source or group IP address and the mfib IPv4 pending entry , use the **debug platform software multicast ip cmfib error**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ip cmfib error {**A.B.C.D** | **pending**}
no debug platform software multicast ip cmfib error {**A.B.C.D** | **pending**}

Syntax Description

A.B.C.D	Specifies the source or group IP address information.
pending	Specifies the mfib IPv4 pending entry error information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast ip cmfib error** command:

PE-3-sp#debug platform software multicast ip cmfib error 232.0.1.4 ver

PE-3-sp#debug platform software multicast ip cmfib error 232.0.1.4 verbose

CMFIB-LC IPv4 verbose error debugging enabled for group 232.0.1.4

PE-3-sp#debug platform software multicast ip cmfib error pending ?

<cr>

PE-3-sp#debug platform software multicast ip cmfib error pending

CMFIB-LC IPv4 error pending debugging enabled

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast ip cmfib event

To display information about source or group IP address, mfib IPv4 ctrl entries events, mfib hw-api events, mfib IPv4 table events, mfib IPv4 pending entry events, and mfib IPv4 table events, use the **debug platform software multicast ip cmfib event**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ip cmfib event {**A.B.C.D** | **ctrl** | **hwapi** | **mdt** | **pending** | **table**}
no debug platform software multicast ip cmfib event {**A.B.C.D** | **ctrl** | **hwapi** | **mdt** | **pending** | **table**}

Syntax Description

A.B.C.D	Specifies the source or group IP address information.
pending	Specifies the mfib IPv4 pending entry information.
ctrl	Specifies the mfib IPv4 ctrl entry events.
hwapi	Specifies the mfib hardware API events.
mdt	Specifies the mfib IPv4 table events.
table	Specifies the mfib IPv4 table events.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast ip cmfib event** command:

```
PE-3-sp#debug platform software multicast ip cmfib event ctrl
CMFIB-LC IPv4 event control debugging enabled
PE-3-sp#debug platform software multicast ip cmfib event hwapi
CMFIB-LC IPv4 event hwapi debugging enabled
PE-3-sp#debug platform software multicast ip cmfib event mdt
CMFIB-LC IPv4 event mdt debugging enabled
PE-3-sp#debug platform software multicast ip cmfib event pending
CMFIB-LC IPv4 event pending debugging enabled
PE-3-sp#debug platform software multicast ip cmfib event table
```

CMFIB-LC IPv4 event table debugging enabled

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast ip hal

To display information about the the multicast hal error, event, timer and packet information, use the **debug platform software multicast ip hal**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ip hal {**error events** | **event events** | **pak** | **timer**}
no debug platform software multicast hal {**error events** | **event events** | **pak** | **timer**}

Syntax Description

event	Specifies the events for the selected group.
error	Specifies the debugging errors.
pak	Specifies the packet information.
timer	Specifies the timer information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast ip hal**command using the event keyword:

```
PE-3-sp#debug platform software multicast ip  hal eve
PE-3-sp#debug platform software multicast ip  hal event
Multicast HAL event log debugging is on
PE-3-sp#
*Oct 30 09:24:48.078 EDT: SP: hal_timer_event: NRPF-AG
*Oct 30 09:24:48.790 EDT: SP: hal_timer_event: S-CHECK
*Oct 30 09:24:49.754 EDT: SP: hal_timer_event: NRPF-AG
*Oct 30 09:24:51.530 EDT: SP: hal_timer_event: NRPF-AG
*Oct 30 09:24:53.298 EDT: SP: hal_timer_event: NRPF-AG
*Oct 30 09:24:55.154 EDT: SP: hal_timer_event: NRPF-AG
```

The following example shows output from the **debug platform software multicast ip hal**command using the error keyword:

```
PE-3-sp#debug platform software multicast ip  hal error
Multicast HAL error log debugging is on
```

The following example shows output from the **debug platform software multicast ip hal**command using the pak keyword:

```
PE-3-sp#debug platform software multicast ip  hal pak
```

```
PE-3-sp#debug platform software multicast ip  hal pak
Multicast HAL packet log debugging is on
```

The following example shows output from the **debug platform software multicast ip hal**command using the timer keyword:

```
PE-3-sp#debug platform software multicast ip  hal tim
PE-3-sp#debug platform software multicast ip  hal timer
Multicast HAL timer log debugging is on
```

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast ipv6

To display information about multicast IPv6 hardware switching, use the **debug platform software multicast ipv6**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ipv6 {**control** | **error group-address** | **event group-address**}
no debug platform software multicast ipv6 {**control** | **error group-address** | **event group-address**}

Syntax Description

control	Displays all multicast hardware switching debugging information, including errors, events, and packets.
error group-address	Displays error messages related to multicast hardware switching for the specified group-address.
event group-address	Displays the run-time sequence of events for multicast hardware switching.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast ipv6**command using the **control** keyword:

```
Router# debug platform software multicast ipv6 control
```

The following example shows output from the **debug platform software multicast ipv6**command using the **error** keyword:

```
Router# debug mls rp ip multicast error
```

The following example shows output from the **debug platform software multicast ipv6**command using the **event** keyword:

```
Router# debug mls rp ip multicast event
```

Related Commands

Command	Description
ipv6 multicast hardware-switching connected	Downloads the interface and mask entry for IPv6 multicast packet.
ipv6 multicast hardware-switching replication-mode ingress	Configures the ingress hardware replication mode for IPv6 multicast packets.

debug platform software multicast ipv6 cmfib

To display information about multicast ipv6 mfib errors, shortcut events, and hardware statistics export information, use the **debug platform software multicast ipv6 cmfib**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ipv6 cmfib {**error** | **event** | **stats**}
no debug platform software multicast ipv6 cmfib {**error** | **event** | **stats**}

Syntax Description

error	Specifies the multicast ipv6 mfib errors.
event	Specifies the mfib IPv4 pending entry information.
stats	Specifies the hardware statistics export information.

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keyword is required.

Examples

The following example shows output from the **debug platform software multicast ipv6 cmfib** command:

PE-3-sp#debug platform software multicast ipv6 cmfib error

CMFIB-LC IPv6 error debugging enabled

PE-3-sp#debug platform software multicast ipv6 cmfib event

CMFIB-LC IPv6 event debugging enabled

PE-3-sp#debug platform software multicast ipv6 cmfib stats

CMFIB-LC IPv6 stats debugging enabled

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast ipv6

To display information about multicast IPv6 hardware switching, use the **debug platform software multicast ipv6**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ipv6 {**control** | **error group-address** | **event group-address**}
no debug platform software multicast ipv6 {**control** | **error group-address** | **event group-address**}

Syntax Description

control	Displays all multicast hardware switching debugging information, including errors, events, and packets.
error group-address	Displays error messages related to multicast hardware switching for the specified group-address.
event group-address	Displays the run-time sequence of events for multicast hardware switching.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast ipv6**command using the **control** keyword:

```
Router# debug platform software multicast ipv6 control
```

The following example shows output from the **debug platform software multicast ipv6**command using the **error** keyword:

```
Router# debug mls rp ip multicast error
```

The following example shows output from the **debug platform software multicast ipv6**command using the **event** keyword:

```
Router# debug mls rp ip multicast event
```

Related Commands

Command	Description
ipv6 multicast hardware-switching connected	Downloads the interface and mask entry for IPv6 multicast packet.
ipv6 multicast hardware-switching replication-mode ingress	Configures the ingress hardware replication mode for IPv6 multicast packets.

debug platform software multicast ipv6 hal

To display information about multicast ipv6 hal errors and event information, use the **debug platform software multicast ipv6 hal**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast ipv6 hal {**error** | **event**}
no debug platform software multicast ipv6 hal {**error** | **event**}

Syntax Description

error	Specifies the multicast ipv6 mfib errors.
event	Specifies the mfib IPv4 pending entry information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keyword is required.

Examples

The following example shows output from the **debug platform software multicast ipv6 hal** command:

PE-3-sp#debug platform software multicast ipv6 hal error

CMFIB-LC IPv6 debugging enabled

PE-3-sp#debug platform software multicast ipv6 hal event

CMFIB-LC IPv6 IPv6 HAL error debugging enabled

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast lc

To display the layer 2 line card multicast events, use the **debug platform software multicast lc**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast lc
no debug platform software multicast lc

Syntax Description

lc	Specifies the line card for which the multicast events are to be displayed.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast lc** command:

```
PE-3-sp#debug platform software multicast lc
Debug from mls_mcast_lc library debugging is on
```

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast mld

To display information about the events and packet information for mld debugging, use the **debug platform software multicast mld**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast mld {**event events** | **pak events**}
debug platform software multicast mld {**event events** | **pak events**}

Syntax Description

event	Specifies the mld events for the selected group.
pak	Specifies the mld packet information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast mld** command using the event keyword:

```
PE-3-sp#debug platform software multicast igmp event
multicast snooping event log debugging is on
```

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast mrouter

To display the multicast router events and packet information, use the **debug platform software multicast mrouter**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast mrouter {**event events** | **pak events**}
no debug platform software multicast mrouter {**event events** | **pak events**}

Syntax Description

event	Specifies the mld events for the selected group.
pak	Specifies the mld packet information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast mrouter** command using the event keyword:

```
PE-3-sp#debug platform software multicast mrouter event
Router Discovery (MLD MROUTER Protocol) event log debugging is on
```

The following example shows output from the **debug platform software multicast mrouter** command using the pak keyword:

```
PE-3-sp#debug platform software multicast mrouter pak
```

Router Discovery (MLD MROUTER Protocol) packet log debugging is on

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast msc

To display information about multicast shortcut debugging, use the **debug platform software multicast msc**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast msc {**error events** | **event events** | **pak events**}
no debug platform software multicast msc {**error events** | **event events** | **pak events**}

Syntax Description

events	Specifies the events for the selected group.
error	Specifies the debugging errors.
pak	Specifies the packet information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast msc** command using the error keyword:

PE-3-sp#debug platform software multicast msc error

Multicast Shortcuts error log debugging is on

The following example shows output from the **debug platform software multicast msc** command using the event keyword:

PE-3-sp#debug platform software multicast msc eve

Multicast Shortcuts event log debugging is on

The following example shows output from the **debug platform software multicast msc** command using the pak keyword:

PE-3-sp#debug platform software multicast msc pak

Multicast Shortcuts packet log debugging is on

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast rgmp

To display information about multicast shortcut debugging, use the **debug platform software multicast rgmp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast rgmp {**event events** | **pak events**}
no debug platform software multicast rgmp {**event events** | **pak events**}

Syntax Description

events	Specifies the events for the selected group.
pak	Specifies the packet information.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast rgmp** command using the event keyword:

PE-3-sp#debug platform software multicast rgmp event

RGMP event log debugging is on

The following example shows output from the **debug platform software multicast rgmp** command using the pak keyword:

PE-3-sp#debug platform software multicast rgmp pak

RGMP packet log debugging is on

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast rpdf

To display information about multicast bidirectional df debugging, use the **debug platform software multicast rpdf**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast rpdf {**error events** | **event events**}
no debug platform software multicast rpdf {**error events** | **event events**}

Syntax Description

events	Specifies the events for the selected group.
error	Specifies the debugging errors.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast rpdf** command using the error keyword:

PE-3-sp#debug platform software multicast rpdf error

Multicast Shortcuts error log debugging is on

The following example shows output from the **debug platform software multicast rpdf** command using the event keyword:

PE-3-sp#debug platform software multicast rpdf eve

Multicast Shortcuts event log debugging is on

The following example shows output from the **debug platform software multicast rpdf** command using the pak keyword:

PE-3-sp#debug platform software multicast rpdf pak

Multicast Shortcuts packet log debugging is on

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software multicast titan

To display information about multicast titan debugging, use the **debug platform software multicast titan**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug platform software multicast titan {**error events** | **event events**}
no debug platform software multicast titan {**error events** | **event events**}

Syntax Description

events	Specifies the events for the selected group.
error	Specifies the debugging errors.

Command Default

Debugging is enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRE	This command was introduced on Cisco 7600 series routers.

Usage Guidelines

Only one of the keywords is required.

Examples

The following example shows output from the **debug platform software multicast titan** command using the error keyword:

PE-3-sp#debug platform software multicast rpdf error

Multicast Bidir RP/DF error log debugging is on

The following example shows output from the **debug platform software multicast titan** command using the event keyword:

PE-3-sp#debug platform software multicast rpdf eve

PE-3-sp#debug platform software multicast rpdf event

Multicast Bidir RP/DF event log debugging is on

Related Commands

Command	Description
debug platform software multicast ha	Displays the high availability multicast shortcuts debugging errors and events.

debug platform software otv

To enable Overlay Transport Virtualization (OTV) debugging on the Cisco IOS daemon (IOSd) Shim layer for OTV-specific forwarding object, use the **debug platform software otv** command in privileged EXEC mode. To disable logging of the debug messages, use the **no** form of this command.

debug platform software otv {**error** | **event** | **packet**}
no debug platform software otv {**error** | **event** | **packet**}

Syntax Description

error	Enables error debugging.
event	Enables event debugging.
packet	Enables packet debugging.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Examples

The following is sample output from the **debug platform software otv** command:

```
Router# debug platform software otv event

*Nov  2 16:49:44.282: FMANRP-OTV: Create decap oce, obj_hdl 448FAD1C, obj_id 8
*Nov  2 16:49:44.283: FMANRP-OTV: efp_id 10 on if_num 26, dpidx 16916300
*Nov  2 16:49:44.283: OTV OCE Message sent for
*Nov  2 16:49:44.284: FMANRP-OTV: Modify decap oce, obj_hdl 448FAD1C, obj_id 8
*Nov  2 16:49:44.284: FMANRP-OTV: efp_id 10 on if_num 26, dpidx 16916300
*Nov  2 16:49:44.284: OTV OCE Message sent for
*Nov  2 16:49:44.284: FMANRP-OTV: Create encap oce, obj_hdl 4D14CE2C obj_id 9
*Nov  2 16:49:44.284: FMANRP-OTV: efp_id 10 on if_num 26, dpidx 16916300
*Nov  2 16:49:44.285: FMANRP-OTV: Next obj hdl 4CB67890 type 1D, obj_id 8E25, type 11
```

Related Commands

Command	Description
show platform software otv fp	Displays the overlay configuration on an OTV edge device on the FMAN on the FP.

debug platform software wccp

To enable Web Cache Control Protocol (WCCP) platform debug messages, use the **debug platform software wccp** command in privileged EXEC mode. To disable WCCP platform debug messages, use the **no** form of this command.

debug platform software wccp {**configuration** | **counters** | **detail** | **messages**}
no debug platform software wccp {**configuration** | **counters** | **detail** | **messages**}

Syntax Description

configuration	Enables configuration related debugs.
counters	Enables statistics collection related debugs.
detail	Enables detailed debugs for all WCCP related configurations.
messages	Enables debugs related to type definition language (TDL) messages being exchanged.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 2.2	This command was introduced.
Cisco IOS XE Release 3.1S	This command was modified. The **counters** keyword was added.

Examples

The following is sample output from the **debug platform software wccp configuration** command:

```
Router# debug platform software wccp configuration
```

A WCCP service is configured:

```
*Jun 17 15:41:04.816: FMANRP-WCCP: Config Service Group (0, 0, 0)
                       acl = , propagate_tos = TRUE, mode_is_closed = FALSE
                       definition_is_valid = TRUE, protocol = 6, priority = 240
                       ass_method = Unknown, fwd_method = Unknown, ret_method = Unknown
                       num_mv_sets = 0, redirection_is_active = FALSE, num_wcs = 0
                       use_source_port = FALSE, ports_defined = TRUE
                       ports[0] = 80
                       ports[1] = 0
                       ports[2] = 0
                       ports[3] = 0
                       ports[4] = 0
                       ports[5] = 0
                       ports[6] = 0
                       ports[7] = 0
*Jun 17 15:41:24.827: FMANRP-WCCP: create ce adjacency: CE = 90.20.1.2, fwd_method = GRE
oce= 0x30692230 adj = 0x306921C0 handle = 0x30692230 obj_id = 135
*Jun 17 15:41:24.827: FMANRP-WCCP: adjacency 90.20.1.2 (4500.0000.0000), router_id 66.66.66.66
 proto=47
*Jun 17 15:41:39.807: FMANRP-WCCP: update mask data, Service Group (0, 0, 0)
                       acl = , propagate_tos = TRUE, mode_is_closed = FALSE
```

```
                        definition_is_valid = TRUE, protocol = 6, priority = 240
                        ass_method = Mask, fwd_method = GRE, ret_method = L2
                        num_mv_sets = 1, redirection_is_active = TRUE, num_wcs = 1
                        use_source_port = FALSE, ports_defined = TRUE
                        wc[0] = 90.20.1.2
                        ports[0] = 80
                        ports[1] = 0
                        ports[2] = 0
                        ports[3] = 0
                        ports[4] = 0
                        ports[5] = 0
                        ports[6] = 0
                        ports[7] = 0
*Jun 17 15:41:39.808: FMANRP-WCCP: Service Group (0, 0, 0) generate merged acl from IOS
*Jun 17 15:41:39.808: FMANRP-WCCP: wccp merged_acl(acl=), p=64 t=64 MCP wccp merged_acl,
num_port=1 result_len=64
```

A WCCP service is configured on an interface:

```
*Jun 17 15:45:17.083: FMANRP-WCCP: Config Service Group (0, 0, 0) to interface
GigabitEthernet0/3/1, direction = IN
*Jun 17 15:45:17.084: FMANRP-WCCP: Attach GigabitEthernet0/3/1 interface info for Service
group (0, 0, 0) if_handle 20, direction Input(0x2)
```

A WCCP service is removed from an interface:

```
*Jun 17 15:46:29.815: FMANRP-WCCP: Unconfig Service Group (0, 0, 0) to interface
GigabitEthernet0/3/1, direction = IN
*Jun 17 15:46:29.815: FMANRP-WCCP: Detach GigabitEthernet0/3/1 interface info for Service
group (0, 0, 0) if_handle 20, direction Input(0x2)
```

A WCCP service group is unconfigured:

```
*Jun 17 15:48:17.224: FMANRP-WCCP: (0 0 0) Delete ce = 90.20.1.2
*Jun 17 15:48:17.225: Failed to retrieve service group params while removing ce
*Jun 17 15:48:17.241: FMANRP-WCCP: Unconfig Service Group (0, 0, 0)
```

The following is sample output from **debug platform software wccp messages**command:

```
Router# debug platform software wccp messages
```

A WCCP service is configured:

```
*Jun 17 15:50:57.796: FMANRP-WCCP: send out (0, 0, 0) wccp_svc_cfg (ADD) to fman-rp
                        pri=0, ce_num=0, ass=Unknown, fwd=Unknown, ret=Unknown
                        protocol=6 use_source_port=0 is_closed=0
                        ports[0] = 80
                        ports[1] = 0
                        ports[2] = 0
                        ports[3] = 0
                        ports[4] = 0
                        ports[5] = 0
                        ports[6] = 0
                        ports[7] = 0
*Jun 17 15:51:14.864: FMANRP-WCCP: send out (0, 0, 0) wccp_ce_cfg (ADD) to fman-rp,
ce=90.20.1.2 ce_id=0.0.0.0 rtr_id=66.66.66.66 fwd_method=GRE obj_id=141
*Jun 17 15:51:29.846: FMANRP-WCCP: send out (0, 0, 0) wccp_svc_cfg (MODIFY) to fman-rp
                        pri=0, ce_num=1, ass=Mask, fwd=GRE, ret=L2
                        protocol=6 use_source_port=0 is_closed=0
                        ports[0] = 80
                        ports[1] = 0
                        ports[2] = 0
```

```
                        ports[3] = 0
                        ports[4] = 0
                        ports[5] = 0
                        ports[6] = 0
                        ports[7] = 0
*Jun 17 15:51:29.847: FMANRP-WCCP: send out (0, 0, 0) wccp_acl_begin to fman-rp
*Jun 17 15:51:29.886: FMANRP-WCCP:   Service Group (0, 0, 0) send out ACL=WCCP_ACL_0x0, 64
 ACEs to fman-rp
*Jun 17 15:51:29.886: FMANRP-WCCP: send out (0, 0, 0) wccp_acl_end to fman-rp
```

A WCCP service is removed from an interface:

```
*Jun 17 15:53:40.710: FMANRP-WCCP: send out (0, 0, 0) wccp_if_svc_bind (ADD) to fman-rp
if_handle=20 dir=IN
```

A WCCP service is removed from an interface:

```
*Jun 17 15:54:36.924: FMANRP-WCCP: send out (0, 0, 0) wccp_if_svc_bind (DELETE) to fman-rp
  if_handle=20 dir=IN
```

A WCCP service group is unconfigured:

```
*Jun 17 15:55:13.117: FMANRP-WCCP: send out (0, 0, 0) wccp_ce_cfg (DELETE) to fman-rp,
ce=90.20.1.2 ce_id=0.0.0.0 rtr_id=0.0.0.0 fwd_method=Unknown obj_id=0
*Jun 17 15:55:13.128: FMANRP-WCCP: send out (0, 0, 0) wccp_svc_cfg (DELETE) to fman-rp
                        pri=0, ce_num=0, ass=Unknown, fwd=Unknown, ret=Unknown
                        protocol=0 use_source_port=0 is_closed=0
                        ports[0] = 0
                        ports[1] = 0
                        ports[2] = 0
                        ports[3] = 0
                        ports[4] = 0
                        ports[5] = 0
                        ports[6] = 0
                        ports[7] = 0
```

The following is sample output from the **debug platform software wccp detail** command:

```
Router# debug platform software wccp detail
```

WCCP service is configured:

```
*Jun 17 18:42:15.491: FMANRP-WCCP: create ce adjacency: CE = 90.20.1.2, fwd_method = GRE
oce= 0x30692230 adj = 0x306921C0 handle = 0x30692230 obj_id = 181
*Jun 17 18:42:30.472: FMANRP-WCCP: Converted adjacency (0x30692230), to ce_addr (90.20.1.2)
*Jun 17 18:42:30.473: FMANRP-WCCP:  Service Group (0, 0, 0) send out ACL=WCCP_ACL_0x0,
ACE=1, obj_id=181 PERMIT, srcopr 5, dstopr 3 to fman-rp
*Jun 17 18:42:30.473: FMANRP-WCCP: oce 0x30692230 adj 0x306921C0 handle 0x30692230
```

The debug messages appear for each access control entry (ACE) of the merged access control list (ACL) for the service group:

```
*Jun 17 18:42:30.487: FMANRP-WCCP: Converted adjacency (0x30692230), to ce_addr (90.20.1.2)
*Jun 17 18:42:30.487: FMANRP-WCCP:  Service Group (0, 0, 0) send out ACL=WCCP_ACL_0x0,
ACE=64, obj_id=181 PERMIT, srcopr 5, dstopr 3 to fman-rp
*Jun 17 18:42:30.487: FMANRP-WCCP: oce 0x30692230 adj 0x306921C0 handle 0x30692230
```

A WCCP service group is unconfigured:

```
*Jun 17 18:46:34.316: FMANRP-WCCP: (0 0 0) Delete ce = 90.20.1.2
*Jun 17 18:46:34.316: Failed to retrieve service group params while removing ce
```

The following is sample output from the **debug platform software wccp counters**command.

```
Router# debug platform software wccp counters
```

Statistics are collected for the first time on a WCCP-enabled interface:

```
*Jun 17 18:50:18.930: FMANRP-WCCP: Received wccp_if_stats intf 20, redirect(IN) 0 from
fman-fp
```

The following debug messages are displayed every 10 seconds:

```
*Jun 17 18:51:18.929: FMANRP-WCCP: Received (0, 0, 0) svc_grp_stats from fman-fp
  unassigned_count = 0, dropped_closed_count = 0
  bypass_count = 0, bypass_failed_count = 0
  denied_count = 0, redirect_count = 0
  num_entries = 0
*Jun 17 18:51:18.929: FMANRP-WCCP: Received wccp_if_stats intf 20, redirect(IN) 0 from
fman-fp
*Jun 17 18:51:28.929: FMANRP-WCCP: Received (0, 0, 0) svc_grp_stats from fman-fp
  unassigned_count = 0, dropped_closed_count = 0
  bypass_count = 0, bypass_failed_count = 0
  denied_count = 0, redirect_count = 0
  num_entries = 0
```

Related Commands

Command	Description
clear ip wccp	Removes WCCP statistics (counts) maintained on the router for a particular service.
ip wccp	Enables support of the specified WCCP service for participation in a service group.
ip wccp check services all	Enables all WCCP services.
ip wccp outbound-acl-check	Enables execution of ACL applied on the actual outgoing interface of a packet before a decision is taken to redirect a packet.
ip wccp redirect	Enables packet redirection on an outbound or inbound interface using WCCP.
show platform software wccp	Displays global statistics related to WCCP on Cisco ASR 1000 Series Routers.

debug pnp

To enable debugging traces in Cisco Open Plug-n-Play (PnP) agent, use the **debug pnp** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug pnp {**all** | **connection** | **discovery** | **infra** | **sasl** | **service** *service-type*}
no debug pnp {**all** | **connection** | **discovery** | **infra** | **sasl** | **service** *service-type*}

Syntax Description

all	Enables all Open Plug-n-Play (PnP) agent debugging.
connection	Enables PnP connection debugging.
discovery	Enables PnP discovery debugging.
infra	Enables PnP infra debugging.
sasl	Enables PnP Simple Authentication and Security Layer (SASL) (used while XMPP authentication) debugging.
service *service-type*	Enables PnP service debugging.

Command Default

Disabled

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.4(2)T	This command was introduced.
Cisco IOS XE Release 3.12S	This command was integrated into Cisco IOS XE Release 3.12S.
15.2(2)E	This command was integrated into Cisco IOS Release 15.2(2)E.

The following example shows how to debug a PnP agent:

```
Device> enable
Device# debug pnp connection
PNP agent connection debugs debugging is on

Device# debug pnp infra
PNP agent infra debugs debugging is on
```

Related Commands

Command	Description
debug xmpp profile	Debugs issues related to PnP agent infrastructure.

debug policy-firewall

Note Effective with Cisco IOS Release 12.4(20)T, the **debug policy-firewall** command replaces the **debug ip inspect** command.

To display messages about Cisco software firewall events, including details about the packets of the protocol, use the **debug policy-firewall** command in priviliged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug policy-firewall {**function-trace** | **object-creation** | **ha** | **object-deletion** | **list** {**access-list** | **extended-access-list**} | **events** | **timers** | **packet-path** | **protocol** *protocol-name* | **L2-transparent** | **control-plane** | **detailed**}
no debug policy-firewall {**function-trace** | **object-creation** | **object-deletion** | **list** {**access-list** | **extended-access-list**} | **events** | **timers** | **packet-path** | **protocol** *protocol-name* | **L2-transparent** | **control-plane** | **detailed** | **ha**}

Syntax Description

function-trace	Displays messages about software functions called by the firewall.
object-creation	Displays messages about software objects being created by the firewall. Object creation corresponds to the beginning of firewall-inspected sessions.
object-deletion	Displays messages about software objects being deleted by the firewall. Object deletion corresponds to the closing of firewall-inspected sessions.
list	Displays messages about policy firewall conditional debugging.
access-list	Filters the basic list of policy firewall conditional debugging messages. The valid range is from 1 to 199.
extended-access-list	Filters the extended range of policy firewall conditional debugging messages. The valid range is from 1300 to 2699.
events	Displays messages about firewall software events, including information about firewall packet processing or MIB special events.
timers	Displays messages about firewall timer events such as when the firewall idle timeout is reached.
packet-path	Displays messages about the packet-path functions.
protocol *protocol-name*	Displays firewall-inspected protocol events. Displays messages about firewall-inspected protocol events, including details about the packets of the protocol.
L2-transparent	Displays messages about Layer 2 transparent (firewall) bridge mode events.

control-plane	Displays messages about the control plane routines.
detailed	Detailed information is displayed for all the other enabled firewall debug commands. Use this form of the command in conjunction with the other firewall **debug** commands.
ha	Displays firewall high availability (HA) log messages.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(20)T	This command was introduced. This command replaces the **debug ip inspect** command.
15.0(1)M	This command was modified. The **list** and **packet-path** keywords were added.
15.2(3)T	This command was modified. The **ha** keyword was added.

Usage Guidelines

The **debug policy-firewall** command is used to troubleshoot firewall problems. You can use the output of this command to analyze the behavior of the firewall and to diagnose the root cause of the problem.

Examples

The following is sample output from the **debug policy-firewall function-trace** command:

```
Device# debug policy-firewall function-trace

Feb 13 08:13:43: FIREWALL: fw_dp_tcp_init_sis():
Feb 13 08:13:43: FIREWALL: fw_dp_insp_init_sis():
Feb 13 08:13:43: FIREWALL: fw_dp_tcp_inspect(): , i2r = 1
Feb 13 08:13:43: FIREWALL: fw_dp_insp_listen_state():
Feb 13 08:13:43: FIREWALL: fw_dp_insp_ensure_return_traffic():
Feb 13 08:13:43: FIREWALL: fw_dp_insp_process_syn_packet():
Feb 13 08:13:43: FIREWALL: fw_dp_insp_create_tcp_host_entry():
Feb 13 08:13:43: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 0
Feb 13 08:13:43: FIREWALL*: fw_dp_insp_synsent_state():
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 1
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_synrcvd_state():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_remove_sis_from_host_entry():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_remove_host_entry():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_delete_host_entry():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_handle_icq_control_stream():
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 0
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_estab_state():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_handle_icq_control_stream():
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 1
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_estab_state():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_handle_icq_control_stream():
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 0
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_estab_state():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_handle_icq_control_stream():
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 1
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_estab_state():
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_handle_icq_control_stream():
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 0
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_estab_state():
```

```
Feb 13 08:13:44: FIREWALL*: fw_dp_tcp_inspect(): , i2r = 1
Feb 13 08:13:44: FIREWALL*: fw_dp_insp_estab_state():
Feb 13 08:13:44: %APPFW-6-IM_ICQ_SESSION: im-icq text-chat service session initiator sends
 77 bytes  session 192.168.3.3:36091 192.168.103.3:5190 on zone-pair zp_test_in class test_im
 appl-class test_icq_1
```

The date in each line of the output is the time stamp. This output shows the functions called by the Cisco IOS firewall as a session is inspected. Entries with an asterisk (*) after the word "FIREWALL" are entries when the fast path is used; otherwise, the process path is used.

The following is sample output from the **debug policy-firewall object-creation**, **debug policy-firewall object-deletion**, **debug policy-firewall timers**, and **debug policy-firewall events** commands:

```
Log Buffer (600000 bytes):
Feb 13 08:16:17: FIREWALL: FW CCE got packet 0x66030694 in process path
Feb 13 08:16:17: FIREWALL: Router gen or router destined pak 0x66030694, let it pass
Feb 13 08:16:17: FIREWALL: FW CCE got packet 0x660311F8 in process path
Feb 13 08:16:17: FIREWALL: Router gen or router destined pak 0x660311F8, let it pass
Feb 13 08:16:17: FIREWALL: FW CCE got packet 0x66030A60 in process path
Feb 13 08:16:17: FIREWALL: Router gen or router destined pak 0x66030A60, let it pass
Feb 13 08:16:19: FIREWALL: FW CCE got packet 0x660328C0 in process path
Feb 13 08:16:19: FIREWALL: Router gen or router destined pak 0x660328C0, let it pass
Feb 13 08:16:21: FIREWALL: FW CCE got packet 0x66031D5C in process path
Feb 13 08:16:21: FIREWALL: Router gen or router destined pak 0x66031D5C, let it pass
Feb 13 08:16:22: FIREWALL: FW CCE got packet 0x66032128 in process path
Feb 13 08:16:22: FIREWALL: Router gen or router destined pak 0x66032128, let it pass
Feb 13 08:16:22: FIREWALL: FW CCE got packet 0x660324F4 in process path
Feb 13 08:16:22: FIREWALL: Router gen or router destined pak 0x660324F4, let it pass
Feb 13 08:16:24: FIREWALL: FW CCE got packet 0x66033424 in process path
Feb 13 08:16:24: FIREWALL: Router gen or router destined pak 0x66033424, let it pass
Feb 13 08:16:25: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:16:25: FIREWALL: fw_dp_insp_sample_session_rate
Feb 13 08:16:26: FIREWALL: FW CCE got packet 0x66032C8C in process path
Feb 13 08:16:26: FIREWALL: Router gen or router destined pak 0x66032C8C, let it pass
Feb 13 08:16:26: FIREWALL: FW CCE got packet 0x6602DCD0 in process path
Feb 13 08:16:26: FIREWALL: Router gen or router destined pak 0x6602DCD0, let it pass
Feb 13 08:16:26: FIREWALL: FW CCE got packet 0x5011DDB4 in process path
Feb 13 08:16:26: FIREWALL: Router gen or router destined pak 0x5011DDB4, let it pass
Feb 13 08:16:28: FIREWALL: FW CCE got packet 0x5011D9E8 in process path
Feb 13 08:16:28: FIREWALL: sis 20491840 : Timer Start: Timer: 20491964 Time: 30000 milisecs
Feb 13 08:16:28: FIREWALL: sis 20491840 : Timer Init Leaf
Feb 13 08:16:28: FIREWALL: sis 20491840 : Allocating L7 sis extensionL4 protocol = 1, L7
protocol = 62, granular = 5
Feb 13 08:16:28: FIREWALL: sis 20491840 : create host entry 669F3180 addr 192.168.103.3
bucket 12 (vrf 0:0) fwfo 0x507E39C0
Feb 13 08:16:29: FIREWALL*: sis 20491840 : Timer Start: Timer: 20491964 Time: 3600000
milisecs
Feb 13 08:16:29: %APPFW-6-IM_ICQ_SESSION: im-icq text-chat service session initiator sends
 77 bytes  session 192.168.3.3:36091 192.168.103.3:5190 on zone-pair zp_test_in class test_im
 appl-class test_icq_1
Feb 13 08:16:29: %APPFW-6-IM_ICQ_SESSION: im-icq text-chat service session initiator gets
198 bytes  session 192.168.103.3:5190 192.168.3.3:36091 on zone-pair zp_test_in class test_im
 appl-class test_icq_1
Feb 13 08:16:29: FIREWALL: FW CCE got packet 0x20159864 in process path
Feb 13 08:16:29: FIREWALL: Router gen or router destined pak 0x20159864, let it pass
Feb 13 08:16:29: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:16:29: FIREWALL: delete host entry 669F3180 addr 192.168.103.3
Feb 13 08:16:30: FIREWALL: FW CCE got packet 0x66033058 in process path
Feb 13 08:16:30: FIREWALL: Router gen or router destined pak 0x66033058, let it pass
Feb 13 08:16:31: FIREWALL: FW CCE got packet 0x660337F0 in process path
Feb 13 08:16:31: FIREWALL: Router gen or router destined pak 0x660337F0, let it pass
Feb 13 08:16:31: FIREWALL: FW CCE got packet 0x20159C30 in process path
```

```
Feb 13 08:16:31: FIREWALL: Router gen or router destined pak 0x20159C30, let it pass
Feb 13 08:16:34: FIREWALL: FW CCE got packet 0x20159FFC in process path
Feb 13 08:16:34: FIREWALL: Router gen or router destined pak 0x20159FFC, let it pass
Feb 13 08:16:35: FIREWALL: FW CCE got packet 0x5011E54C in process path
Feb 13 08:16:35: FIREWALL: Router gen or router destined pak 0x5011E54C, let it pass
Feb 13 08:16:36: FIREWALL: FW CCE got packet 0x665E6304 in process path
Feb 13 08:16:36: FIREWALL: Router gen or router destined pak 0x665E6304, let it pass
Feb 13 08:16:36: FIREWALL: FW CCE got packet 0x5011E180 in process path
Feb 13 08:16:36: FIREWALL: Router gen or router destined pak 0x5011E180, let it pass
Feb 13 08:16:38: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:16:38: FIREWALL: fw_dp_insp_sample_session_rate
Feb 13 08:16:38: FIREWALL: FW CCE got packet 0x2015A3C8 in process path
Feb 13 08:16:38: FIREWALL: Router gen or router destined pak 0x2015A3C8, let it pass
Feb 13 08:16:39: FIREWALL: FW CCE got packet 0x5011E918 in process path
Feb 13 08:16:39: FIREWALL: Router gen or router destined pak 0x5011E918, let it pass
Feb 13 08:16:40: FIREWALL: FW CCE got packet 0x665E6E68 in process path
Feb 13 08:16:40: FIREWALL: Router gen or router destined pak 0x665E6E68, let it pass
Feb 13 08:16:40: FIREWALL: FW CCE got packet 0x2015A794 in process path
Feb 13 08:16:40: FIREWALL: Router gen or router destined pak 0x2015A794, let it pass
Feb 13 08:16:43: FIREWALL: FW CCE got packet 0x665E7234 in process path
Feb 13 08:16:43: FIREWALL: Router gen or router destined pak 0x665E7234, let it pass
Feb 13 08:16:44: FIREWALL: FW CCE got packet 0x5011ECE4 in process path
Feb 13 08:16:44: FIREWALL: Router gen or router destined pak 0x5011ECE4, let it pass
Feb 13 08:16:44: FIREWALL: FW CCE got packet 0x2015AB60 in process path
Feb 13 08:16:44: FIREWALL: Router gen or router destined pak 0x2015AB60, let it pass
Feb 13 08:16:45: FIREWALL: FW CCE got packet 0x665E7600 in process path
Feb 13 08:16:45: FIREWALL: Router gen or router destined pak 0x665E7600, let it pass
Feb 13 08:16:48: FIREWALL: FW CCE got packet 0x665E79CC in process path
Feb 13 08:16:48: FIREWALL: Router gen or router destined pak 0x665E79CC, let it pass
Feb 13 08:16:48: FIREWALL: FW CCE got packet 0x5011F47C in process path
Feb 13 08:16:48: FIREWALL: Router gen or router destined pak 0x5011F47C, let it pass
Feb 13 08:16:49: FIREWALL: FW CCE got packet 0x6602E468 in process path
Feb 13 08:16:49: FIREWALL: Router gen or router destined pak 0x6602E468, let it pass
Feb 13 08:16:50: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:16:50: FIREWALL: fw_dp_insp_sample_session_rate
Feb 13 08:16:50: FIREWALL: FW CCE got packet 0x2015B2F8 in process path
Feb 13 08:16:50: FIREWALL: Router gen or router destined pak 0x2015B2F8, let it pass
Feb 13 08:16:52: FIREWALL: FW CCE got packet 0x6602E09C in process path
Feb 13 08:16:52: FIREWALL: Router gen or router destined pak 0x6602E09C, let it pass
Feb 13 08:16:53: FIREWALL: FW CCE got packet 0x6602EC00 in process path
Feb 13 08:16:53: FIREWALL: Router gen or router destined pak 0x6602EC00, let it pass
Feb 13 08:16:54: FIREWALL: FW CCE got packet 0x6602EFCC in process path
Feb 13 08:16:54: FIREWALL: Router gen or router destined pak 0x6602EFCC, let it pass
Feb 13 08:16:55: FIREWALL: FW CCE got packet 0x6602F764 in process path
Feb 13 08:16:55: FIREWALL: Router gen or router destined pak 0x6602F764, let it pass
Feb 13 08:16:57: FIREWALL: FW CCE got packet 0x6602F398 in process path
Feb 13 08:16:57: FIREWALL: Router gen or router destined pak 0x6602F398, let it pass
Feb 13 08:16:57: FIREWALL: FW CCE got packet 0x6602FB30 in process path
Feb 13 08:16:57: FIREWALL: Router gen or router destined pak 0x6602FB30, let it pass
Feb 13 08:16:59: FIREWALL: FW CCE got packet 0x66030E2C in process path
Feb 13 08:16:59: FIREWALL: Router gen or router destined pak 0x66030E2C, let it pass
Feb 13 08:16:59: FIREWALL: FW CCE got packet 0x66030694 in process path
Feb 13 08:16:59: FIREWALL: Router gen or router destined pak 0x66030694, let it pass
Feb 13 08:17:00: FIREWALL*: sis 20491840 : Timer Start: Timer: 20491964 Time: 5000 milisecs
Feb 13 08:17:00: FIREWALL*: sis 20491840 : Timer Start: Timer: 20491964 Time: 1000 milisecs
Feb 13 08:17:01: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:17:01: FIREWALL: sis 20491840 : Idle Timer Expires: Timer: 20491964
Feb 13 08:17:01: FIREWALL: sis 20491840 : Delete sis half_open 0
Feb 13 08:17:01: FIREWALL: sis 20491840 : Timer Stop: Timer: 20491964
Feb 13 08:17:01: FIREWALL: sis 20491840 : Delete sis
Feb 13 08:17:01: FIREWALL: sis 20491840 : session on temporary delete list
Feb 13 08:17:01: FIREWALL: sis 20491840 : Calling l4 cleanup
Feb 13 08:17:01: FIREWALL: FW CCE got packet 0x660311F8 in process path
Feb 13 08:17:01: FIREWALL: Router gen or router destined pak 0x660311F8, let it pass
```

```
Feb 13 08:17:02: FIREWALL: FW CCE got packet 0x66030A60 in process path
Feb 13 08:17:02: FIREWALL: Router gen or router destined pak 0x66030A60, let it pass
Feb 13 08:17:02: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:17:02: FIREWALL: fw_dp_insp_sample_session_rate
Feb 13 08:17:04: FIREWALL: FW CCE got packet 0x66031990 in process path
Feb 13 08:17:04: FIREWALL: Router gen or router destined pak 0x66031990, let it pass
Feb 13 08:17:04: FIREWALL: FW CCE got packet 0x660315C4 in process path
Feb 13 08:17:04: FIREWALL: Router gen or router destined pak 0x660315C4, let it pass
Feb 13 08:17:06: FIREWALL: FW CCE got packet 0x660328C0 in process path
Feb 13 08:17:06: FIREWALL: Router gen or router destined pak 0x660328C0, let it pass
Feb 13 08:17:07: FIREWALL: FW CCE got packet 0x66031D5C in process path
Feb 13 08:17:07: FIREWALL: Router gen or router destined pak 0x66031D5C, let it pass
Feb 13 08:17:08: FIREWALL: FW CCE got packet 0x66033424 in process path
Feb 13 08:17:08: FIREWALL: Router gen or router destined pak 0x66033424, let it pass
Feb 13 08:17:09: FIREWALL: FW CCE got packet 0x66032C8C in process path
Feb 13 08:17:09: FIREWALL: Router gen or router destined pak 0x66032C8C, let it pass
Feb 13 08:17:11: FIREWALL: FW CCE got packet 0x6602DCD0 in process path
Feb 13 08:17:11: FIREWALL: Router gen or router destined pak 0x6602DCD0, let it pass
Feb 13 08:17:11: FIREWALL: FW CCE got packet 0x5011DDB4 in process path
Feb 13 08:17:11: FIREWALL: Router gen or router destined pak 0x5011DDB4, let it pass
Feb 13 08:17:13: FIREWALL: FW CCE got packet 0x20159498 in process path
Feb 13 08:17:13: FIREWALL: Router gen or router destined pak 0x20159498, let it pass
Feb 13 08:17:13: FIREWALL: FW CCE got packet 0x665E5F38 in process path
Feb 13 08:17:13: FIREWALL: Router gen or router destined pak 0x665E5F38, let it pass
Feb 13 08:17:14: FIREWALL: fw_dp_insp_handle_timer_event
Feb 13 08:17:14: FIREWALL: fw_dp_insp_sample_session_rate
Feb 13 08:17:16: FIREWALL: FW CCE got packet 0x5011D9E8 in process path
Feb 13 08:17:16: FIREWALL: Router gen or router destined pak 0x5011D9E8, let it pass
Feb 13 08:17:16: FIREWALL: FW CCE got packet 0x20159864 in process path
Feb 13 08:17:16: FIREWALL: Router gen or router destined pak 0x20159864, let it pass
```

The following is sample output from the **debug policy-firewall protocol icq** command:

The event debug output declares the packet path from which the firewall got the packet. The packet path can be either Cisco Express Forwarding or the process path. The **debug policy-firewall** command is used when the firewall sends out a packet that acts like a proxy.

The timer debug output specifies timer-related events. Timers are used to close the sessions created by the firewall. Whenever a timeout happens, the timer debugging output specifies whether it needs to close the session or keep it open for longer.

```
Device# debug policy-firewall protocol icq

Apr  2 23:55:21: CCE*: I2R = 1, state_object = 0x0, data_len = 0
Apr  2 23:55:21: CCE*: ICQ protocol found...
Apr  2 23:55:21: CCE*: cce_dp_named_db_inspect_icq_create_cso
Apr  2 23:55:21: CCE*: I2R = 0, state_object = 0x508A1014, data_len = 10
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 1 , Packet length = 4
Apr  2 23:55:21: CCE*: I2R = 1, state_object = 0x508A1014, data_len = 270
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 1 , Packet length = 264
Apr  2 23:55:21: CCE*: ICQ:Find the client version
Apr  2 23:55:21: CCE*: ICQ:Get the client string
Apr  2 23:55:21: CCE*: ICQ:Object Type = 6,Object Length = 256
Apr  2 23:55:21: CCE*: icq_setstate_on_servicetype
Apr  2 23:55:21: CCE*: ICQ:Obj Data Skipping :prev state =4
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 0,Curr state = 1 , Prev state = 0
Apr  2 23:55:21: CCE*: I2R = 0, state_object = 0x508A1014, data_len = 42
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 2 , Packet length = 36
Apr  2 23:55:21: CCE*: ICQ:Family Service Id = 1,Subtype Id = 3
Apr  2 23:55:21: CCE*: ICQ:curr state = 9
```

```
Apr  2 23:55:21: CCE*: I2R = 1, state_object = 0x508A1014, data_len = 56
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 2 , Packet length = 50
Apr  2 23:55:21: CCE*: ICQ:Family Service Id = 1,Subtype Id = 23
Apr  2 23:55:21: CCE*: ICQ:curr state = 22
Apr  2 23:55:21: CCE*: ICQ:service = 1 , version = 4
Apr  2 23:55:21: CCE*: ICQ:service = 19 , version = 4
Apr  2 23:55:21: CCE*: ICQ:service = 2 , version = 1
Apr  2 23:55:21: CCE*: ICQ:service = 3 , version = 1
Apr  2 23:55:21: CCE*: ICQ:service = 21 , version = 1
Apr  2 23:55:21: CCE*: ICQ:Detected ICQ Protocol
Apr  2 23:55:21: CCE*: I2R = 1, state_object = 0x508A1014, data_len = 230
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 2 , Packet length = 224
Apr  2 23:55:21: CCE*: ICQ:Family Service Id = 4,Subtype Id = 6
Apr  2 23:55:21: CCE*: ICQ:curr state = 14
Apr  2 23:55:21: CCE*: icq_process_client_message
Apr  2 23:55:21: CCE*: ICQ:Message Channel ID = 2
Apr  2 23:55:21: CCE*: icq_skip_client_msg
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 5
Apr  2 23:55:21: CCE*: ICQ:length = 190,obj length = 186
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 4,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 3
Apr  2 23:55:21: CCE*: ICQ:length = 0,obj length = 0
Apr  2 23:55:21: CCE*: I2R = 1, state_object = 0x508A1014, data_len = 66
Apr  2 23:55:21: CCE*: ICQ:state = 21
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 2 , Packet length = 60
Apr  2 23:55:21: CCE*: ICQ:Family Service Id = 4,Subtype Id = 6
Apr  2 23:55:21: CCE*: ICQ:curr state = 14
Apr  2 23:55:21: CCE*: icq_process_client_message
Apr  2 23:55:21: CCE*: ICQ:Message Channel ID = 2
Apr  2 23:55:21: CCE*: icq_skip_client_msg
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 5
Apr  2 23:55:21: CCE*: ICQ:length = 26,obj length = 26
Apr  2 23:55:21: CCE*: ICQ:Obj Data Skipping :prev state =19
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 0,Curr state = 1 , Prev state = 0
Apr  2 23:55:21: CCE*: ICQ:service found = 2
Apr  2 23:55:21: CCE*: ICQ: Found IM default service
Apr  2 23:55:21: %APPFW-6-IM_ICQ_SESSION: im-icq un-recognized service session initiator
sends 66 bytes  session 192.168.5.3:25610 63.147.175.30:5190 on zone-pair zp_test_in class
 test_im appl-class test_icq_1
Apr  2 23:55:21: CCE*: I2R = 0, state_object = 0x508A1014, data_len = 36
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 2 , Packet length = 30
Apr  2 23:55:21: CCE*: ICQ:Family Service Id = 4,Subtype Id = 12
Apr  2 23:55:21: CCE*: ICQ:curr state = 9
Apr  2 23:55:21: CCE*: I2R = 0, state_object = 0x508A1014, data_len = 285
Apr  2 23:55:21: CCE*: ICQ:state = 1
Apr  2 23:55:21: CCE*: ICQ:FLAP Channel = 2 , Packet length = 279
Apr  2 23:55:21: CCE*: ICQ:Family Service Id = 4,Subtype Id = 7
Apr  2 23:55:21: CCE*: ICQ:curr state = 14
Apr  2 23:55:21: CCE*: icq_process_client_message
Apr  2 23:55:21: CCE*: ICQ:Message Channel ID = 2
Apr  2 23:55:21: CCE*: icq_skip_client_msg
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 1
Apr  2 23:55:21: CCE*: ICQ:length = 241,obj length = 2
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 239,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 6
Apr  2 23:55:21: CCE*: ICQ:length = 235,obj length = 4
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 231,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 5
Apr  2 23:55:21: CCE*: ICQ:length = 227,obj length = 4
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 223,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 15
```

```
Apr  2 23:55:21: CCE*: ICQ:length = 219,obj length = 4
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 215,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 3
Apr  2 23:55:21: CCE*: ICQ:length = 211,obj length = 4
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 207,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 5
Apr  2 23:55:21: CCE*: ICQ:length = 203,obj length = 190
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 13,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 22
Apr  2 23:55:21: CCE*: ICQ:length = 9,obj length = 4
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 5,Curr state = 19 , Prev state = 19
Apr  2 23:55:21: CCE*: ICQ:TLV Service Type = 19
Apr  2 23:55:21: CCE*: ICQ:length = 1,obj length = 1
Apr  2 23:55:21: CCE*: ICQ:Obj Data Skipping :prev state =19
Apr  2 23:55:21: CCE*: ICQ:ICQ Data length = 0,Curr state = 1 , Prev state = 0
Apr  2 23:56:10: CCE*: I2R = 1, state_object = 0x508A1014, data_len = 0
Apr  2 23:56:11: FIREWALL sis 65A1C100: Sis extension deleted
Apr  2 23:56:11: CCE: cce_dp_named_db_inspect_icq_delete_cso
```

The sample output from the **debug policy-firewall protocol winmsgr** command includes information about the instant messenger (IM) service. For example, the following lines declare that the type of IM service the user is running is Windows Messenger (WINMSGR):

The debug output details the different states that the state machine sees while parsing the Layer 7 I Seek You (ICQ) payload.

```
Apr  3 00:21:46: CCE*: WINMSGR:service found = 2
Apr  3 00:21:46: CCE*: WINMSGR: Found IM default service
```

The following is sample output from the **debug policy-firewall protocol winmsgr** command:

```
Device# debug policy-firewall protocol winmsgr

Apr  3 00:21:46: CCE*: I2R = 1, state_object = 0x0, data_len = 0
Apr  3 00:21:46: CCE*: WINMSGR protocol found...
Apr  3 00:21:46: CCE*: cce_dp_named_db_inspect_winmsgr_create_cso
Apr  3 00:21:46: CCE*: I2R = 1, state_object = 0x660CF5B4, data_len = 19
Apr  3 00:21:46: CCE*: WINMSGR:datalen=19,matchflag=11,matchlen=19
Apr  3 00:21:46: CCE*: WINMSGR:Initial trafficfound
Apr  3 00:21:46: CCE*: I2R = 0, state_object = 0x660CF5B4, data_len = 19
Apr  3 00:21:46: CCE*: WINMSGR:datalen=19,matchflag=11,matchlen=19
Apr  3 00:21:46: CCE*: WINMSGR:Initial trafficfound
Apr  3 00:21:46: CCE*: I2R = 1, state_object = 0x660CF5B4, data_len = 82
Apr  3 00:21:46: CCE*: WINMSGR:datalen=82,matchflag=6,matchlen=4
Apr  3 00:21:46: CCE*: WINMSGR:version msg : CVR 31 0x0409 winnt 5.0 i386 MSMSGS 5.1.0701
WindowsMessenger fwuser@example.com
Apr  3 00:21:46: CCE*: I2R = 0, state_object = 0x660CF5B4, data_len = 96
Apr  3 00:21:46: CCE*: WINMSGR:datalen=96,matchflag=6,matchlen=4
Apr  3 00:21:46: CCE*: I2R = 1, state_object = 0x660CF5B4, data_len = 33
Apr  3 00:21:46: CCE*: WINMSGR:datalen=33,matchflag=12,matchlen=33
Apr  3 00:21:46: CCE*: WINMSGR:Initial trafficfound
Apr  3 00:21:46: CCE*: I2R = 0, state_object = 0x660CF5B4, data_len = 162
Apr  3 00:21:46: CCE*: I2R = 1, state_object = 0x660CF5B4, data_len = 324
Apr  3 00:21:46: CCE*: I2R = 0, state_object = 0x660CF5B4, data_len = 37
Apr  3 00:21:46: CCE*: WINMSGR:datalen=37,matchflag=12,matchlen=37
Apr  3 00:21:46: CCE*: WINMSGR:Initial trafficfound
Apr  3 00:21:46: CCE*: I2R = 1, state_object = 0x660CF5B4, data_len = 307
Apr  3 00:21:46: CCE*: WINMSGR:datalen=307,matchflag=5,matchlen=118
Apr  3 00:21:46: CCE*: WINMSGR:service found = 2
Apr  3 00:21:46: CCE*: WINMSGR: Found IM default service
Apr  3 00:21:46: %APPFW-6-IM_WINMSGR_SESSION: im-winmsgr un-recognized service session
initiator sends 307 bytes  session 192.168.5.3:24601 209.165.200.230:1863 on zone-pair
```

```
zp_test_in class test_im appl-class test_winmsgr_1
Apr  3 00:21:46: CCE*: I2R = 0, state_object = 0x660CF5B4, data_len = 320
Apr  3 00:21:46: CCE*: I2R = 0, state_object = 0x660CF5B4, data_len = 332
Apr  3 00:21:46: CCE*: WINMSGR:datalen=332,matchflag=5,matchlen=143
Apr  3 00:21:46: CCE*: WINMSGR:service found = 2
Apr  3 00:21:46: CCE*: WINMSGR: Found IM default service
Apr  3 00:21:46: %APPFW-6-IM_WINMSGR_SESSION: im-winmsgr un-recognized service session
initiator gets 332 bytes  session 209.165.200.230:1863 192.168.5.3:24601 on zone-pair
zp_test_in class test_im appl-class test_winmsgr_1
Apr  3 00:23:11: CCE*: I2R = 1, state_object = 0x660CF5B4, data_len = 0
Apr  3 00:23:11: FIREWALL sis 65A1D540: Sis extension deleted
```

The following is sample output from the **debug policy-firewall control-plane** command:

```
Device# debug policy-firewall control-plane

policy_fw:
  Policy-Firewall control-plane debugging is on
voice-gw-118.03#
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
No Active Message Discriminator.
No Inactive Message Discriminator.
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 247 messages logged, xml disabled,
                     filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 44 message lines logged
Log Buffer (60000000 bytes):
FIREWALL CP: fw_cp_prot_num_to_name()  l4 1, l7 5, gran 0
FIREWALL CP: fw_cp_get_flow_policy_and_class()  Flow policy does not exist
FIREWALL CP: fw_cp_check_create_default_l7_policy()  Could not retrieve flow policy for L4
 policy l4-pmap L4 class l4-cmap
FIREWALL CP: fw_classmap_filter_update_in_policymap()  Adding filter 0x650187F0 to class
l4-cmap in policy l4-pmap
FIREWALL CP: fw_policy_action_cmd()  PPM create action inspect with params 0x64CAF8E8
FIREWALL CP: fw_inspect_class_params()  inspect config-plane CLASS-ADD action
0x66315C5C,params 0x64CAF8E8
FIREWALL CP: fw_validate_class_for_matchprot()  Validating protocols in class l4-cmap
FIREWALL CP: fw_validate_class_for_matchprot()  protocol filter found
FIREWALL CP: fw_inspect_class_params()  Attached config-plane action_params 0x663BD280
FIREWALL CP: fw_cp_create_attach_flow_policy()
FIREWALL CP: fw_cp_get_string_from_random_num()  Random number generated is 2697258553
FIREWALL CP: fw_cp_generate_random_string()  Allocated random str 2697258553 for policy
l4-pmap class l4-cmap
FIREWALL CP: fw_cp_get_random_string()  Found random string  for policy l4-pmap class l4-cmap
FIREWALL CP: fw_cp_get_random_string()  Found random string  for policy l4-pmap class l4-cmap
FIREWALL CP: fw_cp_get_random_string()  Found random string  for policy l4-pmap class l4-cmap
FIREWALL CP: fw_cp_prot_num_to_name()  l4 2, l7 5, gran 0
FIREWALL CP: fw_inspect_int_class_params()
FIREWALL CP: fw_create_attach_template_class()
FIREWALL CP: fw_create_attach_template_class()  Creating template class for trigger
15udp_2697258553 in 15_2697258553
FIREWALL CP: fw_create_attach_template_class()  Trying to create a PPM filter with id
0x64CA73EC
FIREWALL CP: fw_cp_prot_num_to_name()  l4 4, l7 5, gran 0
FIREWALL CP: fw_inspect_int_class_params()
FIREWALL CP: fw_create_attach_template_class()
FIREWALL CP: fw_create_attach_template_class()  Creating template class for trigger
15icmp_2697258553 in 15_2697258553
```

```
FIREWALL CP: fw_create_attach_template_class()  Trying to create a PPM filter with id
0x64CA73EC
FIREWALL CP: fw_cp_create_attach_vtcp_classes()  Create policy 15
FIREWALL CP: fw_cp_create_tcp_15()
FIREWALL CP: fw_cp_vtcp_support_get_tcp_init_class()  Creating TCP Class with Pure SYN
filter
FIREWALL CP: fw_inspect_int_class_params()
FIREWALL CP: fw_create_attach_template_class()
FIREWALL CP: fw_create_attach_template_class()  Creating template class for trigger
15tcp_2697258553 in 15_2697258553
FIREWALL CP: fw_create_attach_template_class()  Trying to create a PPM filter with id
0x64CA73A4
FIREWALL CP: fw_cp_create_attach_flow_policy()  Success-creating flow policy
FIREWALL CP: fw_cp_create_attach_flow_policy()  Attach flow policy to trigger class as child
 policy
FIREWALL CP: fw_cp_create_attach_flow_policy()  Success- Attached flow policy to trigger
class
FIREWALL CP: fw_cp_create_attach_flow_policy()  Creating P20 & P21 for vtcp
FIREWALL CP: fw_cp_generate_random_string()  Found random string  for policy l4-pmap class
 l4-cmap
FIREWALL CP: fw_cp_get_flow_policy_and_class()  Found flow policy 0x64FFC838
FIREWALL CP: fw_cp_get_random_string()  Found random string  for policy l4-pmap class l4-cmap
FIREWALL CP: fw_cp_get_random_string()  Found random string  for policy l4-pmap class l4-cmap
FIREWALL CP: fw_cp_get_flow_policy_and_class()  Found flow TCP 0x6585718C and UDP 0x645D1794
 classes
FIREWALL CP: fw_cp_check_create_default_l7_class()  Checking the class l4-cmap
FIREWALL CP: fw_reverse_policy_handle_zp_event()
FIREWALL CP: fw_reverse_policy_handle_zp_event()  Reverse_policy Zone pair add event
FIREWALL CP: fw_get_ppm_policy_on_zp()  Did not find ppm policy on zp zp p_type 0x7
FIREWALL CP: fw_get_name_type_and_client_of_first_class_in_policy()
FIREWALL CP: fw_create_cp_dynamic_class()
FIREWALL CP: fw_create_cp_dynamic_class()  Trying to create a PPM filter with id 0x10000000
FIREWALL CP: fw_create_cp_dynamic_class()  Success
FIREWALL CP: fw_drop_class_params()  action 0x6637A5C0, cmd_params 0x64CA7550, event 0x21
FIREWALL CP: fw_create_noop_feature_object()
FIREWALL CP: fw_create_inspect_feature_object()
FIREWALL CP: fw_create_fo_internal()  Create FO for class 0xC0000002 target_class 0xA0000000
 action CCE_INSPECT_CONFIGURED
FIREWALL CP: fw_cp_get_inspect_params()
FIREWALL CP: fw_cp_get_inspect_params()  Creating the FO with default parameters
FIREWALL CP: fw_create_fo_internal()  Created FO with id 0xAAAA0006 action
CCE_INSPECT_CONFIGURED
FIREWALL CP: fw_cp_store_fo_id()  Enqueue  0xAAAA0006 to fo_param_list
FIREWALL CP: fw_create_noop_feature_object()
FIREWALL CP: fw_create_inspect_int_feature_object()
FIREWALL CP: fw_create_fo_internal()  Create FO for class 0xC0000005 target_class 0xA0000000
 action CCE_INSPECT
FIREWALL CP: fw_cp_get_inspect_params()
FIREWALL CP: fw_cp_get_inspect_params()  Creating the FO with default parameters
FIREWALL CP: fw_create_fo_internal()  Created FO with id 0xAAAA0007 action CCE_INSPECT
FIREWALL CP: fw_cp_store_fo_id()  Enqueue  0xAAAA0007 to fo_param_list
FIREWALL CP: fw_create_noop_feature_object()
FIREWALL CP: fw_create_inspect_int_feature_object()
FIREWALL CP: fw_create_fo_internal()  Create FO for class 0xC0000007 target_class 0xA0000000
 action CCE_INSPECT
FIREWALL CP: fw_cp_get_inspect_params()
FIREWALL CP: fw_cp_get_inspect_params()  Creating the FO with default parameters
FIREWALL CP: fw_create_fo_internal()  Created FO with id 0xAAAA0008 action CCE_INSPECT
FIREWALL CP: fw_cp_store_fo_id()  Enqueue  0xAAAA0008 to fo_param_list
FIREWALL CP: fw_create_noop_feature_object()
FIREWALL CP: fw_create_inspect_int_feature_object()
FIREWALL CP: fw_create_fo_internal()  Create FO for class 0xC0000009 target_class 0xA0000000
 action CCE_INSPECT
FIREWALL CP: fw_cp_get_inspect_params()
```

```
FIREWALL CP: fw_cp_get_inspect_params()  Creating the FO with default parameters
FIREWALL CP: fw_create_fo_internal()  Created FO with id 0xAAAA0009 action CCE_INSPECT
FIREWALL CP: fw_cp_store_fo_id()  Enqueue  0xAAAA0009 to fo_param_list
FIREWALL CP: fw_create_drop_feature_object()
FIREWALL CP: fw_create_fo_internal()  Create FO for class 0xC0000003 target_class 0xA0000000
 action CCE_FW_DROP
FIREWALL CP: fw_create_fo_internal()  Created FO with id 0xAAAA000A action CCE_FW_DROP
FIREWALL CP: fw_create_internal_reverse_policy()
FIREWALL CP: fw_create_ppm_reverse_policy()
FIREWALL CP: fw_get_name_type_and_client_of_first_class_in_policy()
FIREWALL CP: fw_create_cp_dynamic_class()
FIREWALL CP: fw_create_noop_feature_object()
FIREWALL CP: fw_create_noop_feature_object()
%SYS-5-CONFIG_I: Configured from console by console
FIREWALL CP: fw_cp_prot_num_to_name()  l4 1, l7 5, gran 0
FIREWALL CP: fw_drop_class_params()  action 0x6637A5C0, cmd_params 0x00000000, event 0x40
FIREWALL CP: fw_get_ppm_policy_on_zp()  Found ppm policy l4-pmap on zp zp p_type 0x7
```

The following is sample output from the **debug policy-firewall L2-transparent** command:

```
Device# debug policy-firewall L2-transparent

*Apr  4 08:28:23.554: L2FW*:insp_l2_fast_inspection: pak 673DBD90, input-interface
FastEthernet1/1, output-interface FastEthernet1/0
*Apr  4 08:28:23.554: L2FW*:Src 17.3.39.1 dst 17.3.39.3 protocol tcp
*Apr  4 08:28:23.554: TBAP: Check AuthProxy is configured on idb=FastEthernet1/1 path=1
linktype=38
*Apr  4 08:28:23.554: L2FW:Input ACL not configured or the ACL is bypassed
*Apr  4 08:28:23.554: L2FW:Output ACL is not configured or ACL is bypassed
*Apr  4 08:28:23.554: L2FW*:IP inspect firewall is not cfged on  input or output
interface.PASS
*Apr  4 08:28:23.554: L2FW* 2:insp_l2_fast_inspection: pak 673DBD90, input-interface
FastEthernet1/1, output-interface FastEthernet1/0
*Apr  4 08:28:23.554:  CCE L2 FW
*Apr  4 08:28:23.554: L2FW* -3:insp_l2_fast_inspection: pak 673DBD90, input-interface
FastEthernet1/1, output-interface FastEthernet1/0
```

The following is sample output from the **debug policy-firewall detailed** command:

```
Device# debug policy-firewall detailed

Log Buffer (600000 bytes):
Feb 13 08:40:01: FIREWALL: ret_val 0 is not FW_DP_INSP_PASS_PAK
<snip>
Feb 13 08:41:22: FIREWALL: ret_val 0 is not FW_DP_INSP_PASS_PAK
Feb 13 08:41:24: FIREWALL: ret_val 0 is not FW_DP_INSP_PASS_PAK
Feb 13 08:41:25: FIREWALL*: Searching for FSO in class 0x50793C20class group 0x10000000,
target 0x1, cce class type 0x2B
Feb 13 08:41:25: FIREWALL*: not found
Feb 13 08:41:25: FIREWALL*: Try to create session in fastpath
Feb 13 08:41:25: FIREWALL: Searching for FSO in class 0x50793C20class group 0x10000000,
target 0x1, cce class type 0x2B
Feb 13 08:41:25: FIREWALL: not found
Feb 13 08:41:25: FIREWALL: Create FSO
Feb 13 08:41:25: FIREWALL: sis 204925C0 : fw_dp_state_object_link
Feb 13 08:41:25: FIREWALL: sis 204925C0 : FO class 0x50793C20 class group 0x10000000, target
 0x1, FO 0x20255D80
Feb 13 08:41:25: FIREWALL: sis 204925C0 : alert = 1, audit_trail = 0
Feb 13 08:41:25: FIREWALL: sis 204925C0 : l7 protocol 62, granular = 5
Feb 13 08:41:25: FIREWALL: sis 204925C0 : fw_dp_state_object_attach_forward
Feb 13 08:41:25: FIREWALL: sis 204925C0 : fw_dp_state_object_create_and_attach_reverse
Feb 13 08:41:25: FIREWALL: sis 204925C0 : FSO bind success for reverse class 0x50793C80class
 group 0x10000000, target 0x1
```

```
Feb 13 08:41:25: FIREWALL: sis 204925C0 :Session Info :
Feb 13 08:41:25: session->fwfo 0x507E39C0
Feb 13 08:41:25: class type 0x2B, target 0x1, policy id 0x10000000, class id 0x50793C20
Feb 13 08:41:25: class type 0x2B, reverse target 0x1, reverse policy id 0x10000000, reverse
 class id 0x50793C80
Feb 13 08:41:25: src addr 192.168.3.3, port 36091, vrf id 0
Feb 13 08:41:25: dst addr 192.168.103.3, port 5190, vrf id 0
Feb 13 08:41:25: L4 Protocol : TCP
Feb 13 08:41:25: FIREWALL: sis 204925C0 : L4 inspection returned 3
Feb 13 08:41:25: FIREWALL*: FSO feature object 0x204925C0 found
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : L4 inspection returned 3
Feb 13 08:41:25: FIREWALL*: FSO feature object 0x204925C0 found
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : max_sessions 2147483647; current sessions 0
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : IM : Token set for L7 named-db
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : cce_sb 0x66A5BA00, pak 0x50028974, data_len 0
in_fast_path 1, dir = 1
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : p_app_data = C174268, p_data_len = 6p_offset =
0
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : Found particle offset token, data1 = 0
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : Opening 0 channels for icq
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : icq L7 inspect result: PASS packet
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : L4 inspection returned 3
Feb 13 08:41:25: FIREWALL*: FSO feature object 0x204925C0 found
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : cce_sb 0x66A5BA00, pak 0x5004CAC8, data_len 10
in_fast_path 1, dir = 2
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : p_app_data = C210848, p_data_len = Ap_offset =
0
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : Found particle offset token, data1 = 0
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : Opening 0 channels for icq
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : icq L7 inspect result: PASS packet
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : L4 inspection returned 3
Feb 13 08:41:25: FIREWALL*: FSO feature object 0x204925C0 found
Feb 13 08:41:25: FIREWALL*: sis 204925C0 : cce_sb 0x66A5BA00, pak 0x50028974, data_len 270
 in_fast_path 1, dir = 1
```

The following is sample output from the **debug policy-firewall ha** command

```
Device# debug policy-firewall ha

*May 19 14:17:19.991: FIREWALL: IOS FW RF stat event: status: RF_STATUS_PEER_COMM
my state: STANDBY HOT peer state: ACTIVE
*May 19 14:17:19.995: FIREWALL: IOS FW RF stat event: status: RF_STATUS_PEER_PRESENCE
my state: STANDBY HOT peer state: DISABLED
*May 19 14:17:19.995: FIREWALL: RG with ID:1 state STANDBY: found
*May 19 14:17:19.995: FIREWALL: Event for RG-1: RF_PROG_ACTIVE_FAST
*May 19 14:17:19.995: FIREWALL: RG with ID:1 state ACTIVE: found
*May 19 14:17:19.995: FIREWALL: Standbyhot to Active transition for RG 1
*May 19 14:17:19.995: FIREWALL sis 30CEEF40: Timer Start: Timer: 30CEEFD4 Time: 30000 ms
*May 19 14:17:19.995: FIREWALL: RG 1 trasitioned to Active
*May 19 14:17:19.995: FIREWALL: RG with ID:1 state ACTIVE: found
*May 19 14:17:19.995: FIREWALL: RG with ID:1 state ACTIVE: found
*May 19 14:17:19.995: FIREWALL: RG with ID:1 state ACTIVE: found
May 19 14:17:30.003: FIREWALL: Event for RG-1: RF_PROG_STANDBY_BULK Configuring Zone Based
 Firewall Redundancy Draft Copy Cisco systems, Inc. Company Confidential
*May 19 14:17:30.003: FIREWALL: ret_val 0 is not PASS_PAK
*May 19 14:17:30.003: FIREWALL: RG with ID:1 state ACTIVE: found
*May 19 14:17:30.003: FIREWALL: Starting BulkSync for RG 1
*May 19 14:17:30.003: FIREWALL sis 30CEEF40: Bulk sync session 30CEEF40 needs to be failed
 over(add)
*May 19 14:17:30.003: FIREWALL: ret_val 0 is not PASS_PAK
*May 19 14:17:30.003: FIREWALL sis 30CEEF40: Send add session message
(192.168.7.205:32424:0)=>(192.168.107.1:23:0) l4_prot tcp
*May 19 14:17:30.003: FIREWALL: BulkSync done; Send BulkEnd
```

debug policy-firewall exporter

To log NetFlow Version 9 debug messages, use the **debug policy-firewall exporter** command in privileged EXEC mode.

debug policy-firewall exporter

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.4(2)T	This command was introduced.

Usage Guidelines

Use this command to troubleshoot NetFlow Version 9 flow exporter issues.

Examples

The following is sample output from the **debug policy-firewall exporter** command:

```
Device# debug policy-firewall exporter

Policy-Firewall NetFlow Logging debugging is on

Feb 10 04:00:44.899 EST: FW-EXPORT: [process] FnF registration start
Feb 10 04:00:44.899 EST: FW-EXPORT: [init] data template (0) initialized successfully
Feb 10 04:00:44.903 EST: FW-EXPORT: [init] data template (1) initialized successfully
Feb 10 04:00:44.903 EST: FW-EXPORT: [init] data template (2) initialized successfully
Feb 10 04:00:44.903 EST: FW-EXPORT: [init] data template (3) initialized successfully
Feb 10 04:00:44.903 EST: FW-EXPORT: [init] data template (4) initialized successfully
Feb 10 04:00:44.903 EST: FW-EXPORT: [init] data template (5) initialized successfully
Feb 10 04:00:45.499 EST: FW-EXPORT: Option template (Class-Table) registration successful
Feb 10 04:00:45.499 EST: FW-EXPORT: Sent Optional Record class id:(0x0) <--> Name:(UNKNOWN)
Feb 10 04:00:45.499 EST: FW-EXPORT: Sent Optional Record class id:(0x456A941) <-->
Name:(netflow_cm)
Feb 10 04:00:45.499 EST: FW-EXPORT: Sent Optional Record class id:(0x639) <-->
Name:(class-default)
Feb 10 04:00:45.827 EST: FW-EXPORT: Option template (Protocol-Table) registration successful
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000000) <-->
Name:(Unknown)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000001) <-->
Name:(ftp)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000002) <-->
Name:(telnet)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000003) <-->
Name:(smtp)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000004) <-->
Name:(http)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000005) <-->
Name:(tacacs)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000006) <-->
Name:(dns)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000007) <-->
Name:(sql-net)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000008) <-->
Name:(https)
```

```
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000009) <-->
Name:(tftp)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600000A) <-->
Name:(gopher)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600000B) <-->
Name:(finger)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600000C) <-->
Name:(kerberos)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600000D) <-->
Name:(pop2)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600000E) <-->
Name:(pop3)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600000F) <-->
Name:(sunrpc)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000010) <-->
Name:(msrpc)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000011) <-->
Name:(nntp)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000012) <-->
Name:(snmp)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000013) <-->
Name:(imap)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000014) <-->
Name:(ldap)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000015) <-->
Name:(exec)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000016) <-->
Name:(login)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000017) <-->
Name:(shell)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000018) <-->
Name:(ms-sql)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000019) <-->
Name:(sybase-sql)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600001A) <-->
Name:(nfs)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600001B) <-->
Name:(lotusnote)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600001C) <-->
Name:(h323)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600001D) <-->
Name:(h323-annexe)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600001E) <-->
Name:(h323-nxg)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x600001F) <-->
Name:(cuseeme)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000020) <-->
Name:(realmedia)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000021) <-->
Name:(netshow)
Feb 10 04:00:45.827 EST: FW-EXPORT: Sent Optional Record Protocol id:(0x6000022) <-->
Name:(streamworks)
!
!
!
```

Related Commands

Command	Description
flow exporter	Creates or modifies a Flexible NetFlow flow exporter and enters flow exporter configuration mode.
show flow exporter	Displays Flexible NetFlow flow exporter status and statistics.

debug policy-firewall mib

To toggle on or off the support for MIBs in a zone-based policy firewall, use the **debug policy-firewall mib** command in privileged EXEC mode. To disable the MIB support, use the **no** form of this command.

debug policy-firewall mib {**event** | **object-creation** | **object-deletion** | **object-retrieval**}
no debug policy-firewall mib {**event** | **object-creation** | **object-deletion** | **object-retrieval**}

Syntax Description

event	Turns on debugging for a firewall MIB event.
object-creation	Turns on debugging for a firewall MIB object creation.
object-deletion	Turns on debugging for a firewall MIB object deletion.
object-retrieval	Turns on debugging for a firewall MIB object retrieval.

Command Default

Privileged EXEC (#)

Command History

Release	Modification
15.1(1)T	This command was introduced.

Usage Guidelines

This command provides debug support for MIBs in zone-based policy firewall similar to the Cisco IOS firewall.

Examples

The following is a sample output from the **debug policy-firewall mib object-retrieval** command:

```
Router# debug policy-firewall mib object-retrieval
Firewall MIB object retrieval debugging is on
```

debug port-channel load-balance

To enable debug output for port-channel load balancing, use the **debug port-channel load-balance** command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug port-channel load-balance {**all** | **manual** | **weighted**}
no debug port-channel load-balance {**all** | **manual** | **weighted**}

Syntax Description

all	Turns on debugging for all load-balancing operations.
manual	Turns on debugging for only manual load-balancing operations.
weighted	Turns on debugging for only weighted load-balancing operations.

Command Default

Port-channel debugging is turned off.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)S	This command was introduced.

Usage Guidelines

Use this command to help troubleshoot load balancing of service instances over port-channel member links.

Examples

The following example shows how to enable debugging for only weighted load-balancing operations:

```
Router# debug port-channel load-balance weighted
Port-channel Load-Balance Weighted debugging is on
```

debug pots

To display information on the telephone interfaces, use the **debug pots**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug pots {**driver** | **csm**} [{**1** | **2**}]
no debug pots {**driver** | **csm**} [{**1** | **2**}]

Syntax Description

driver	Displays driver debug information.
csm	Displays Content Switching Module (CSM) debug information.
1	(Optional) Displays information for telephone port 1 only.
2	(Optional) Displays information for telephone port 2 only.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug pots** command displays driver and CSM debug information for telephone ports 1 and 2.

Examples

The following is sample output from the **debug pots driver 1** command. This sample display indicates that the telephone port driver is not receiving caller ID information from the ISDN line. Therefore, the analog caller ID device attached to the telephone port does not display caller ID information.

```
Router# debug pots driver 1
00:01:51:POTS DRIVER port=1 activate ringer: cadence=0 callerId=Unknown
00:01:51:POTS DRIVER port=1 state=Idle drv_event=RING_EVENT
00:01:51:POTS DRIVER port=1 enter_ringing
00:01:51:POTS DRIVER port=1 cmd=19
00:01:51:POTS DRIVER port=1 activate disconnect
00:01:51:POTS DRIVER port=1 state=Ringing drv_event=DISCONNECT_EVENT
00:01:51:POTS DRIVER port=1 cmd=1A
00:01:51:POTS DRIVER port=1 enter_idle
00:01:51:POTS DRIVER port=1 ts connect: 0 0
00:01:51:POTS DRIVER port=1 cmd=D
00:01:51:POTS DRIVER port=1 report onhook
00:01:51:POTS DRIVER port=1 activate tone=SILENCE_TONE
00:01:51:POTS DRIVER port=1 state=Idle drv_event=TONE_EVENT
00:01:51:POTS DRIVER port=1 activate tone=SILENCE_TONE
00:01:51:POTS DRIVER port=1 state=Idle drv_event=TONE_EVENT
00:01:53:POTS DRIVER port=1 activate ringer: cadence=0 callerId=Unknown
00:01:53:POTS DRIVER port=1 state=Idle drv_event=RING_EVENT
00:01:53:POTS DRIVER port=1 enter_ringing
00:01:53:POTS DRIVER port=1 cmd=19
00:01:55:POTS DRIVER port=1 cmd=1A
00:02:49:POTS DRIVER port=1 state=Ringing drv_event=OFFHOOK_EVENT
00:02:49:POTS DRIVER port=1 cmd=1A
00:02:49:POTS DRIVER port=1 enter_suspend
00:02:49:POTS DRIVER port=1 cmd=A
00:02:49:POTS DRIVER port=1 report offhook
00:02:49:POTS DRIVER port=1 activate connect: endpt=1 calltype=TWO_PARTY_CALL
00:02:49:POTS DRIVER port=1 state=Suspend drv_event=CONNECT_EVENT
00:02:49:POTS DRIVER port=1 enter_connect: endpt=1 calltype=0
00:02:49:POTS DRIVER port=1 cmd=A
00:02:49:POTS DRIVER port=1 ts connect: 1 0
```

```
00:02:49:POTS DRIVER port=1 activate connect: endpt=1 calltype=TWO_PARTY_CALL
00:02:49:POTS DRIVER port=1 state=Connect drv_event=CONNECT_EVENT
00:02:49:POTS DRIVER port=1 enter_connect: endpt=1 calltype=0
00:02:49:POTS DRIVER port=1 cmd=A
00:02:49:POTS DRIVER port=1 ts connect: 1 0
00:02:55:POTS DRIVER port=1 state=Connect drv_event=ONHOOK_EVENT
00:02:55:POTS DRIVER port=1 enter_idle
00:02:55:POTS DRIVER port=1 ts connect: 0 0
00:02:55:POTS DRIVER port=1 cmd=D
00:02:55:POTS DRIVER port=1 report onhook
00:02:55:POTS DRIVER port=1 activate tone=SILENCE_TONE
00:02:55:POTS DRIVER port=1 state=Idle drv_event=TONE_EVENT
00:02:55:POTS DRIVER port=1 activate tone=SILENCE_TONE
00:02:55:POTS DRIVER port=1 state=Idle drv_event=TONE_EVENT
```

The following is sample output from the **debug pots csm 1** command. This sample display indicates that a dial peer contains an invalid destination pattern (555-1111).

```
Router# debug pots csm 1
01:57:28:EVENT_FROM_ISDN:dchanidb=0x66CB38, call_id=0x11, ces=0x2 bchan=0x0, event=0x1,
cause=0x0
01:57:28:Dial peer not found, route call to port 1
01:57:28:CSM_PROC_IDLE:CSM_EVENT_ISDN_CALL, call_id=0x11, port=1
01:57:28:Calling number '5551111'
01:57:40:CSM_PROC_RINGING:CSM_EVENT_VDEV_OFFHOOK, call_id=0x11, port=1
01:57:40:EVENT_FROM_ISDN:dchan_idb=0x66CB38, call_id=0x11, ces=0x2 bchan=0x0, event=0x4,
cause=0x0
01:57:40:CSM_PROC_CONNECTING:CSM_EVENT_ISDN_CONNECTED, call_id=0x11, port=1
01:57:47:CSM_PROC_CONNECTING:CSM_EVENT_VDEV_ONHOOK, call_id=0x11, port=1
01:57:201863503872: %ISDN-6-DISCONNECT:Interface BRI0:1 disconnected from unknown, call
lasted 5485 seconds
01:57:47: %ISDN-6-DISCONNECT:Interface BRI0:1 disconnected from unknown, call lasted 5485
seconds
01:57:47:EVENT_FROM_ISDN:dchan _idb=0x66CB38, call_id=0x11, ces=0x2 bchan=0xFFFFFFFF,
event=0x0, cause=0x1
01:57:47:CSM_PROC_NEAR_END_DISCONNECT:CSM_
```

debug pots csm

To activate events from which an application can determine and display the status and progress of calls to and from plain old telephone service (POTS) ports, use the **debugpotscsm**command in privileged EXEC mode.

debug pots csm

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1.(2)XF	This command was introduced on the Cisco 800 series routers.

Examples

To see debugging messages, enter the **loggingconsole**global configuration mode command as follows:

```
Router(config)# logging console
Router(config)# exit
```

Debugging messages are displayed in one of two formats that are relevant to the POTS dial feature:

```
hh:mm:ss: CSM_STATE: CSM_EVENT, call id = ??, port = ?
```

or

```
hh:mm:ss: EVENT_FROM_ISDN:dchan_idb=0x???????, call_id=0x????, ces=? bchan=0x????????,
event=0x?, cause=0x??
```

The following table describes the significant fields shown in the display.

Table 65: debug pots csm Field Descriptions

Command Elements	Description
hh:mm:ss	Timestamp (in hours, minutes, and seconds).
CSM_STATE	One of the call CSM states listed in the field description table.
CSM_EVENT	One of the CSM events listed in the field description table.
call id	Hexadecimal value from 0x00 to 0xFF.
port	Telephone port 1 or 2.
EVENT_FROM_ISDN	A CSM event. The table shows a list of CSM events.
dchan_idb	Internal data structure address.
ces	Connection end point suffix used by ISDN.

Command Elements	Description
bchan	Channel used by the call. A value of 0xFFFFFFFF indicates that a channel is not assigned.
event	A hexadecimal value that is translated into a CSM event. The field description table shows a list of events and the corresponding CSM events.
cause	A hexadecimal value that is given to call-progressing events. The field description table shows a list of cause values and definitions.

The following table shows the values for CSM states.

Table 66: CSM States

CSM State	Description
CSM_IDLE_STATE	Telephone on the hook.
CSM_RINGING	Telephone ringing.
CSM_SETUP	Setup for outgoing call in progress.
CSM_DIALING	Dialing number of outgoing call.
CSM_IVR_DIALING	Interactive voice response (IVR) for Japanese telephone dialing.
CSM_CONNECTING	Waiting for carrier to connect the call.
CSM_CONNECTED	Call connected.
CSM_DISCONNECTING	Waiting for carrier to disconnect the call.
CSM_NEAR_END_DISCONNECTING	Waiting for carrier to disconnect the call.
CSM_HARD_HOLD	Call on hard hold.
CSM_CONSULTATION_HOLD	Call on consultation hold.
CSM_WAIT_FOR_HOLD	Waiting for carrier to put call on hard hold.
CSM_WAIT_FOR_CONSULTATION_HOLD	Waiting for carrier to put call on consultation hold.
CSM_CONFERENCE	Waiting for carrier to complete call conference.
CSM_TRANSFER	Waiting for carrier to transfer call.
CSM_APPLIC_DIALING	Call initiated from Cisco IOS command-line interface (CLI).

The following table shows the values for CSM events.

Table 67: CSM Events

CSM Events	Description
CSM_EVENT_INTER_DIGIT_TIMEOUT	Time waiting for dial digits has expired.
CSM_EVENT_TIMEOUT	Near- or far-end disconnect timeout.
CSM_EVENT_ISDN_CALL	Incoming call.
CSM_EVENT_ISDN_CONNECTED	Call connected.
CSM_EVENT_ISDN_DISCONNECT	Far end disconnected.
CSM_EVENT_ISDN_DISCONNECTED	Call disconnected.
CSM_EVENT_ISDN_SETUP	Outgoing call requested.
CSM_EVENT_ISDN_SETUP_ACK	Outgoing call accepted.
CSM_EVENT_ISDN_PROC	Call proceeding and dialing completed.
CSM_EVENT_ISDN_CALL_PROGRESSING	Call being received in band tone.
CSM_EVENT_ISDN_HARD_HOLD	Call on hard hold.
CSM_EVENT_ISDN_HARD_HOLD_REJ	Hold attempt rejected.
CSM_EVENT_ISDN_CHOLD	Call on consultation hold.
CSM_EVENT_ISDN_CHOLD_REJ	Consultation hold attempt rejected.
CSM_EVENT_ISDN_RETRIEVED	Call retrieved.
CSM_EVENT_ISDN_RETRIEVE_REJ	Call retrieval attempt rejected.
CSM_EVENT_ISDN_TRANSFERRED	Call transferred.
CSM_EVENT_ISDN_TRANSFER_REJ	Call transfer attempt rejected.
CSM_EVENT_ISDN_CONFERENCE	Call conference started.
CSM_EVENT_ISDN_CONFERENCE_REJ	Call conference attempt rejected.
CSM_EVENT_ISDN_IF_DOWN	ISDN interface down.
CSM_EVENT_ISDN_INFORMATION	ISDN information element received (used by NTT IVR application).
CSM_EVENT_VDEV_OFFHOOK	Telephone off the hook.
CSM_EVENT_VDEV_ONHOOK	Telephone on the hook.
CSM_EVENT_VDEV_FLASHHOOK	Telephone hook switch has flashed.
CSM_EVENT_VDEV_DIGIT	DTMF digit has been detected.

CSM Events	Description
CSM_EVENT_VDEV_APPLICATION_CALL	Call initiated from Cisco IOS CLI.

The following table shows the values for events that are translated into CSM events.

Table 68: Event Values

Hexadecimal Value	Event	CSM Event
0x0	DEV_IDLE	CSM_EVENT_ISDN_DISCONNECTED
0x1	DEV_INCALL	CSM_EVENT_ISDN_CALL
0x2	DEV_SETUP_ACK	CSM_EVENT_ISDN_SETUP_ACK
0x3	DEV_CALL_PROC	CSM_EVENT_ISDN_PROC
0x4	DEV_CONNECTED	CSM_EVENT_ISDN_CONNECTED
0x5	DEV_CALL_PROGRESSING	CSM_EVENT_ISDN_CALL_PROGRESSING
0x6	DEV_HOLD_ACK	CSM_EVENT_ISDN_HARD_HOLD
0x7	DEV_HOLD_REJECT	CSM_EVENT_ISDN_HARD_HOLD_REJ
0x8	DEV_CHOLD_ACK	CSM_EVENT_ISDN_CHOLD
0x9	DEV_CHOLD_REJECT	CSM_EVENT_ISDN_CHOLD_REJ
0xa	DEV_RETRIEVE_ACK	CSM_EVENT_ISDN_RETRIEVED
0xb	DEV_RETRIEVE_REJECT	CSM_EVENT_ISDN_RETRIEVE_REJ
0xc	DEV_CONFR_ACK	CSM_EVENT_ISDN_CONFERENCE
0xd	DEV_CONFR_REJECT	CSM_EVENT_ISDN_CONFERENCE_REJ
0xe	DEV_TRANS_ACK	CSM_EVENT_ISDN_TRANSFERRED
0xf	DEV_TRANS_REJECT	CSM_EVENT_ISDN_TRANSFER_REJ

The following table shows cause values that are assigned only to call-progressing events.

Table 69: Cause Values

Hexadecimal Value	Cause Definitions
0x01	UNASSIGNED_NUMBER
0x02	NO_ROUTE
0x03	NO_ROUTE_DEST
0x04	NO_PREFIX

Hexadecimal Value	Cause Definitions
0x06	CHANNEL_UNACCEPTABLE
0x07	CALL_AWARDED
0x08	CALL_PROC_OR_ERROR
0x09	PREFIX_DIALED_ERROR
0x0a	PREFIX_NOT_DIALED
0x0b	EXCESSIVE_DIGITS
0x0d	SERVICE_DENIED
0x10	NORMAL_CLEARING
0x11	USER_BUSY
0x12	NO_USER_RESPONDING
0x13	NO_USER_ANSWER
0x15	CALL_REJECTED
0x16	NUMBER_CHANGED
0x1a	NON_SELECTED_CLEARING
0x1b	DEST_OUT_OF_ORDER
0x1c	INVALID_NUMBER_FORMAT
0x1d	FACILITY_REJECTED
0x1e	RESP_TO_STAT_ENQ
0x1f	UNSPECIFIED_CAUSE
0x22	NO_CIRCUIT_AVAILABLE
0x26	NETWORK_OUT_OF_ORDER
0x29	TEMPORARY_FAILURE
0x2a	NETWORK_CONGESTION
0x2b	ACCESS_INFO_DISCARDED
0x2c	REQ_CHANNEL_NOT_AVAIL
0x2d	PRE_EMPTED
0x2f	RESOURCES_UNAVAILABLE
0x32	FACILITY_NOT_SUBSCRIBED

Hexadecimal Value	Cause Definitions
0x33	BEARER_CAP_INCOMPAT
0x34	OUTGOING_CALL_BARRED
0x36	INCOMING_CALL_BARRED
0x39	BEARER_CAP_NOT_AUTH
0x3a	BEAR_CAP_NOT_AVAIL
0x3b	CALL_RESTRICTION
0x3c	REJECTED_TERMINAL
0x3e	SERVICE_NOT_ALLOWED
0x3f	SERVICE_NOT_AVAIL
0x41	CAP_NOT_IMPLEMENTED
0x42	CHAN_NOT_IMPLEMENTED
0x45	FACILITY_NOT_IMPLEMENT
0x46	BEARER_CAP_RESTRICTED
0x4f	SERV_OPT_NOT_IMPLEMENT
0x51	INVALID_CALL_REF
0x52	CHAN_DOES_NOT_EXIST
0x53	SUSPENDED_CALL_EXISTS
0x54	NO_CALL_SUSPENDED
0x55	CALL_ID_IN_USE
0x56	CALL_ID_CLEARED
0x58	INCOMPATIBLE_DEST
0x5a	SEGMENTATION_ERROR
0x5b	INVALID_TRANSIT_NETWORK
0x5c	CS_PARAMETER_NOT_VALID
0x5f	INVALID_MSG_UNSPEC
0x60	MANDATORY_IE_MISSING
0x61	NONEXISTENT_MSG
0x62	WRONG_MESSAGE

Hexadecimal Value	Cause Definitions
0x63	BAD_INFO_ELEM
0x64	INVALID_ELEM_CONTENTS
0x65	WRONG_MSG_FOR_STATE
0x66	TIMER_EXPIRY
0x67	MANDATORY_IE_LEN_ERR
0x6f	PROTOCOL_ERROR
0x7f	INTERWORKING_UNSPEC

Examples

This section provides debug output examples for three call scenarios, displaying the sequence of events that occur during a POTS dial call or POTS disconnect call.

Call Scenario 1

In this example call scenario, port 1 is on the hook, the application dial is set to call 4085552221, and the far-end successfully connects.

```
Router# debug pots csm
Router# test pots 1 dial 4085552221#
Router#
```

The following output shows an event indicating that port 1 is being used by the dial application:

```
01:58:27: CSM_PROC_IDLE: CSM_EVENT_VDEV_APPLICATION_CALL, call id = 0x0, port = 1
```

The following output shows events indicating that the CSM is receiving the application digits of the number to dial:

```
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:58:27: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
```

The following output shows that the telephone connected to port 1 is off the hook:

```
01:58:39: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_OFFHOOK, call id = 0x0, port = 1
```

The following output shows a call-proceeding event pair indicating that the router ISDN software has sent the dialed digits to the ISDN switch:

```
01:58:40: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8004, ces=0x1 bchan=0x0, event=0x3,
```

```
 cause=0x0
01:58:40: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_ISDN_PROC, call id =
0x8004, port = 1
```

The following output shows the call-progressing event pair indicating that the telephone at the far end is ringing:

```
01:58:40: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8004, ces=0x1 bchan=0xFFFFFFFF,
event=0x5, cause=0x0
01:58:40: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_ISDN_CALL_PROGRESSING, call id = 0x8004, port
 = 1
```

The following output shows a call-connecting event pair indicating that the telephone at the far end has answered:

```
01:58:48: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8004, ces=0x1 bchan=0xFFFFFFFF,
event=0x4, cause=0x0
01:58:48: CSM_PROC_CONNECTING: CSM_EVENT_ISDN_CONNECTED, call id = 0x8004, port = 1
```

The following output shows a call-progressing event pair indicating that the telephone at the far end has hung up and that the calling telephone is receiving an in-band tone from the ISDN switch:

```
01:58:55: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8004, ces=0x1 bchan=0xFFFFFFFF,
event=0x5, cause=0x10
01:58:55: CSM_PROC_CONNECTED: CSM_EVENT_ISDN_CALL_PROGRESSING, call id = 0x8004, port = 1
```

The following output shows that the telephone connected to port 1 has hung up:

```
01:58:57: CSM_PROC_CONNECTED: CSM_EVENT_VDEV_ONHOOK, call id = 0x8004, port = 1
```

The following output shows an event pair indicating that the call has been terminated:

```
01:58:57: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8004, ces=0x1 bchan=0xFFFFFFFF,
event=0x0, cause=0x0
01:58:57: CSM_PROC_NEAR_END_DISCONNECT: CSM_EVENT_ISDN_DISCONNECTED, call id = 0x8004, port
 = 1
813_local#
```

Call Scenario 2

In this example scenario, port 1 is on the hook, the application dial is set to call 4085552221, and the destination number is busy.

```
Router# debug pots csm
Router# test pots 1 dial 4085552221#
Router#
```

The following output shows that port 1 is used by the dial application:

```
01:59:42: CSM_PROC_IDLE: CSM_EVENT_VDEV_APPLICATION_CALL, call id = 0x0, port = 1
```

The following output shows the events indicating that the CSM is receiving the application digits of the number to call:

```
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
```

```
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
01:59:42: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
```

The following output shows an event indicating that the telephone connected to port 1 is off the hook:

```
01:59:52: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_OFFHOOK, call id = 0x0, port = 1
```

The following output shows a call-proceeding event pair indicating that the telephone at the far end is busy:

```
01:59:52: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8005, ces=0x1 bchan=0x0, event=0x3,
 cause=0x11
01:59:52: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_ISDN_PROC, call id = 0x8005, port = 1
```

The following output shows a call-progressing event pair indicating that the calling telephone is receiving an in-band busy tone from the ISDN switch:

```
01:59:58: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8005, ces=0x1 bchan=0xFFFFFFFF,
event=0x5, cause=0x0
01:59:58: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_ISDN_CALL_PROGRESSING, call id = 0x8005, port
 = 1
```

The following output shows an event indicating that the calling telephone has hung up:

```
02:00:05: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_VDEV_ONHOOK, call id = 0x8005, port = 1
```

The following output shows an event pair indicating that the call has been terminated:

```
02:00:05: EVENT_FROM_ISDN:dchan_idb=0x280AF38, call_id=0x8005, ces=0x1 bchan=0xFFFFFFFF,
event=0x0, cause=0x0
02:00:05: CSM_PROC_NEAR_END_DISCONNECT: CSM_EVENT_ISDN_DISCONNECTED, call id = 0x8005, port
 = 1
```

Call Scenario 3

In this example call scenario, port 1 is on the hook, the application dial is set to call 4086661112, the far end successfully connects, and the command **testpotsdisconnect** terminates the call:

```
Router# debug pots csm
Router# test pots 1 dial 4086661112
Router#
```

The following output follows the same sequence of events as shown in Call Scenario 1:

```
1d03h: CSM_PROC_IDLE: CSM_EVENT_VDEV_APPLICATION_CALL, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
```

```
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_DIGIT, call id = 0x0, port = 1
1d03h: CSM_PROC_APPLIC_DIALING: CSM_EVENT_VDEV_OFFHOOK, call id = 0x0, port = 1
1d03h: EVENT_FROM_ISDN:dchan_idb=0x2821F38, call_id=0x8039, ces=0x1
   bchan=0x0, event=0x3, cause=0x0
1d03h: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_ISDN_PROC, call id = 0x8039, port = 1
1d03h: EVENT_FROM_ISDN:dchan_idb=0x2821F38, call_id=0x8039, ces=0x1
   bchan=0xFFFFFFFF, event=0x5, cause=0x0
1d03h: CSM_PROC_ENBLOC_DIALING: CSM_EVENT_ISDN_CALL_PROGRESSING, call id = 0x8039,     port
 = 1
Router# test pots 1 disconnect
```

The **testpotsdisconnect** command disconnects the call before you physically need to put the telephone back on the hook:

```
1d03h: CSM_PROC_CONNECTING: CSM_EVENT_VDEV_APPLICATION_HANGUP_CALL, call id = 0x8039,
 port = 1
1d03h: EVENT_FROM_ISDN:dchan_idb=0x2821F38, call_id=0x8039, ces=0x1
   bchan=0xFFFFFFFF, event=0x0, cause=0x0
1d03h: CSM_PROC_DISCONNECTING: CSM_EVENT_ISDN_DISCONNECTED, call id = 0x8039,
    port = 1
1d03h: CSM_PROC_DISCONNECTING: CSM_EVENT_TIMEOUT, call id = 0x8039, port = 1
```

debug ppp

To display information on traffic and exchanges in an internetwork implementing the Point-to-Point Protocol (PPP), use the **debug ppp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ppp {**packet** | **negotiation** | **error** | **authentication** | **compression** | **cbcp**}
no debug ppp {**packet** | **negotiation** | **error** | **authentication** | **compression** | **cbcp**}

Syntax Description

packet	Displays PPP packets being sent and received. (This command displays low-level packet dumps.)
negotiation	Displays PPP packets sent during PPP startup, where PPP options are negotiated.
error	Displays protocol errors and error statistics associated with PPP connection negotiation and operation.
authentication	Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.
compression	Displays information specific to the exchange of PPP connections using Microsoft Point-to-Point Compression (MPPC). This command is useful for obtaining incorrect packet sequence number information where MPPC compression is enabled.
cbcp	Displays protocol errors and statistics associated with PPP connection negotiations using Microsoft Callback (MSCB).

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug ppp** command when trying to find the following:

- The Network Control Protocols (NCPs) that are supported on either end of a PPP connection
- Any loops that might exist in a PPP internetwork
- Nodes that are (or are not) properly negotiating PPP connections
- Errors that have occurred over the PPP connection
- Causes for CHAP session failures
- Causes for PAP session failures
- Information specific to the exchange of PPP connections using the Callback Control Protocol (CBCP), used by Microsoft clients
- Incorrect packet sequence number information where MPPC compression is enabled

Refer to Internet RFCs 1331, 1332, and 1333 for details concerning PPP-related nomenclature and protocol information.

Caution The **debug ppp compression** command is CPU-intensive and should be used with caution. This command should be disabled immediately after debugging.

Examples

The following is sample output from the **debug ppp packet** command as seen from the Link Quality Monitor (LQM) side of the connection. This example depicts packet exchanges under normal PPP operation.

```
Router# debug ppp packet
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 3 (C) magic D3454
PPP Serial4: input(C021) state = OPEN code = ECHOREQ(9) id = 3 len = 12
PPP Serial4: O LCP ECHOREP(A) id 3 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 4 (C) magic D3454
PPP Serial4: input(C021) state = OPEN code = ECHOREQ(9) id = 4 len = 12
PPP Serial4: O LCP ECHOREP(A) id 4 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 5 (C) magic D3454
PPP Serial4: input(C021) state = OPEN code = ECHOREQ(9) id = 5 len = 12
PPP Serial4: O LCP ECHOREP(A) id 5 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 6 (C) magic D3454
PPP Serial4: input(C021) state = OPEN code = ECHOREQ(9) id = 6 len = 12
PPP Serial4: O LCP ECHOREP(A) id 6 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 7 (C) magic D3454
PPP Serial4: input(C021) state = OPEN code = ECHOREQ(9) id = 7 len = 12
PPP Serial4: O LCP ECHOREP(A) id 7 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
```

The following table describes the significant fields shown in the display.

Table 70: debug ppp packet Field Descriptions

Field	Description
PPP	PPP debugging output.
Serial4	Interface number associated with this debugging information.
(o), O	Packet was detected as an output packet.

Field	Description
(i), I	Packet was detected as an input packet.
lcp_slqr()	Procedure name; running LQM, send a Link Quality Report (LQR).
lcp_rlqr()	Procedure name; running LQM, received an LQR.
input (C021)	Router received a packet of the specified packet type (in hexadecimal notation). A value of C025 indicates packet of type LQM.
state = OPEN	PPP state; normal state is OPEN.
magic = D21B4	Magic Number for indicated node; when output is indicated, this is the Magic Number of the node on which debugging is enabled. The actual Magic Number depends on whether the packet detected is indicated as I or O.
datagramsize 52	Packet length including header.
code = ECHOREQ(9)	Identifies the type of packet received. Both forms of the packet, string and hexadecimal, are presented.
len = 48	Packet length without header.
id = 3	ID number per Link Control Protocol (LCP) packet format.
pkt type 0xC025	Packet type in hexadecimal notation; typical packet types are C025 for LQM and C021 for LCP.
LCP ECHOREQ(9)	Echo Request; value in parentheses is the hexadecimal representation of the LCP type.
LCP ECHOREP(A)	Echo Reply; value in parentheses is the hexadecimal representation of the LCP type.

To elaborate on the displayed output, consider the partial exchange. This sequence shows that one side is using ECHO for its keepalives and the other side is using LQRs.

```
Router# debug ppp packet
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 3 (C) magic D3454
PPP Serial4: input(C021) state = OPEN code = ECHOREQ(9) id = 3 len = 12
PPP Serial4: O LCP ECHOREP(A) id 3 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
```

The first line states that the router with debugging enabled has sent an LQR to the other side of the PPP connection:

```
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
```

The next two lines indicate that the router has received a packet of type C025 (LQM) and provides details about the packet:

```
PPP Serial4(i): pkt type 0xC025, datagramsize 52
PPP Serial4(i): lcp_rlqr() state = OPEN magic = D3454, len = 48
```

The next two lines indicate that the router received an ECHOREQ of type C021 (LCP). The other side is sending ECHOs. The router on which debugging is configured for LQM but also responds to ECHOs.

```
PPP Serial4(i): pkt type 0xC021, datagramsize 16
PPP Serial4: I LCP ECHOREQ(9) id 3 (C) magic D3454
```

Next, the router is detected to have responded to the ECHOREQ with an ECHOREP and is preparing to send out an LQR:

```
PPP Serial4: O LCP ECHOREP(A) id 3 (C) magic D21B4
PPP Serial4(o): lcp_slqr() state = OPEN magic = D21B4, len = 48
```

The following is sample output from the **debug ppp negotiation** command. This is a normal negotiation, where both sides agree on Network Control Program (NCP) parameters. In this case, protocol type IP is proposed and acknowledged.

```
Router# debug ppp negotiation
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 3D56CAC
ppp: received config for type = 4 (QUALITYTYPE) acked
ppp: received config for type = 5 (MAGICNUMBER) value = 3D567F8 acked (ok)
PPP Serial4: state = ACKSENT fsm_rconfack(C021): rcvd id 5
ppp: config ACK received, type = 4 (CI_QUALITYTYPE), value = C025
ppp: config ACK received, type = 5 (CI_MAGICNUMBER), value = 3D56CAC
ppp: ipcp_reqci: returning CONFACK.
 (ok)
PPP Serial4: state = ACKSENT fsm_rconfack(8021): rcvd id 4
```

The following table describes significant fields shown in the display.

Table 71: debug ppp negotiation Field Descriptions

Field	Description
ppp	PPP debugging output.
sending CONFREQ	Router sent a configuration request.
type = 4 (CI_QUALITYTYPE)	Type of LCP configuration option that is being negotiated and a descriptor. A type value of 4 indicates Quality Protocol negotiation; a type value of 5 indicates Magic Number negotiation.
value = C025/3E8	For Quality Protocol negotiation, indicates NCP type and reporting period. In the example, C025 indicates LQM; 3E8 is a hexadecimal value translating to about 10 seconds (in hundredths of a second).
value = 3D56CAC	For Magic Number negotiation, indicates the Magic Number being negotiated.
received config	Receiving node has received the proposed option negotiation for the indicated option type.
acked	Acknowledgment and acceptance of options.

Field	Description
state = ACKSENT	Specific PPP state in the negotiation process.
ipcp_reqci	IPCP notification message; sending CONFACK.
fsm_rconfack (8021)	Procedure fsm_rconfack processes received CONFACKs, and the protocol (8021) is IP.

The first two lines indicate that the router is trying to bring up LCP and will use the indicated negotiation options (Quality Protocol and Magic Number). The value fields are the values of the options themselves. C025/3E8 translates to Quality Protocol LQM. 3E8 is the reporting period (in hundredths of a second). 3D56CAC is the value of the Magic Number for the router.

```
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 3D56CAC
```

The next two lines indicate that the other side negotiated for options 4 and 5 as requested and acknowledged both. If the responding end does not support the options, a CONFREJ is sent by the responding node. If the responding end does not accept the value of the option, a Configure-Negative-Acknowledge (CONFNAK) is sent with the value field modified.

```
ppp: received config for type = 4 (QUALITYTYPE) acked
ppp: received config for type = 5 (MAGICNUMBER) value = 3D567F8 acked (ok)
```

The next three lines indicate that the router received a CONFAK from the responding side and displays accepted option values. Use the rcvd id field to verify that the CONFREQ and CONFACK have the same ID field.

```
PPP Serial4: state = ACKSENT fsm_rconfack(C021): rcvd id 5
ppp: config ACK received, type = 4 (CI_QUALITYTYPE), value = C025
ppp: config ACK received, type = 5 (CI_MAGICNUMBER), value = 3D56CAC
```

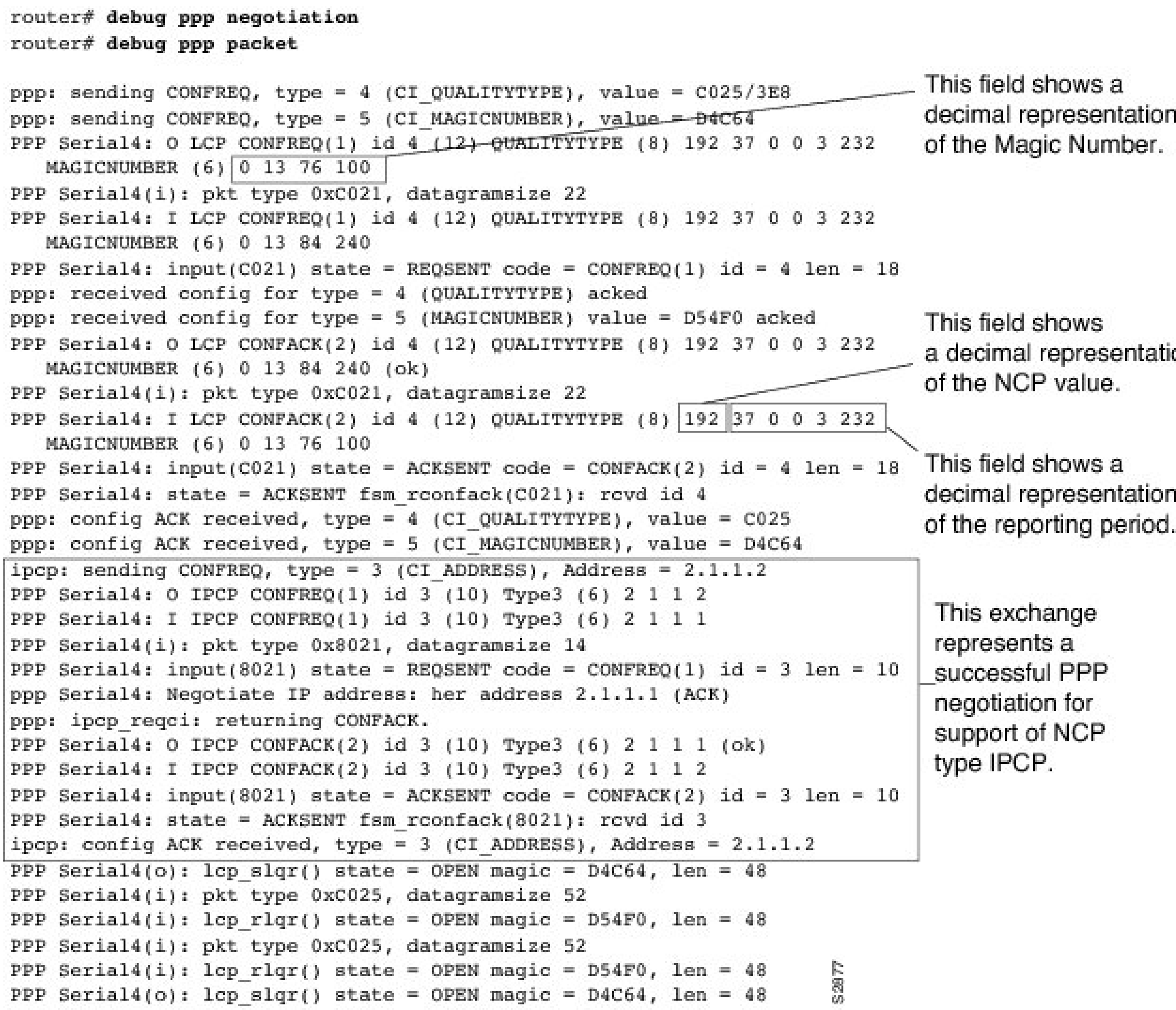
The next line indicates that the router has IP routing enabled on this interface and that the IPCP NCP negotiated successfully:

```
ppp: ipcp_reqci: returning CONFACK.
```

In the last line, the state of the router is listed as ACKSENT.

```
PPP Serial4: state = ACKSENT fsm_rconfack(C021): rcvd id 5\
```

The following is sample output from when the **debug ppp packet** and **debug ppp negotiation** commands are enabled at the same time.



The following is sample output from the **debug ppp negotiation** command when the remote side of the connection is unable to respond to LQM requests:

```
Router# debug ppp negotiation
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
```

```
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44B7010
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44C1488
```

The following is sample output when no response is detected for configuration requests (with both the **debug ppp negotiation** and **debug ppp packet** commands enabled):

```
Router# debug ppp negotiation
Router# debug ppp packet
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44DFDC8
PPP Serial4: O LCP CONFREQ(1) id 14 (12) QUALITYTYPE (8) 192 37 0 0 3 232
   MAGICNUMBER (6) 4 77 253 200
ppp: TIMEout: Time= 44E0980 State= 3
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44DFDC8
PPP Serial4: O LCP CONFREQ(1) id 15 (12) QUALITYTYPE (8) 192 37 0 0 3 232
   MAGICNUMBER (6) 4 77 253 200
ppp: TIMEout: Time= 44E1828 State= 3
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44DFDC8
PPP Serial4: O LCP CONFREQ(1) id 16 (12) QUALITYTYPE (8) 192 37 0 0 3 232
   MAGICNUMBER (6) 4 77 253 200
ppp: TIMEout: Time= 44E27C8 State= 3
ppp: sending CONFREQ, type = 4 (CI_QUALITYTYPE), value = C025/3E8
ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 44DFDC8
PPP Serial4: O LCP CONFREQ(1) id 17 (12) QUALITYTYPE (8) 192 37 0 0 3 232
   MAGICNUMBER (6) 4 77 253 200
ppp: TIMEout: Time= 44E3768 State= 3
```

The following is sample output from the **debug ppp error** command. These messages might appear when the Quality Protocol option is enabled on an interface that is already running PPP.

```
Router# debug ppp error
PPP Serial3(i): rlqr receive failure. successes = 15
PPP: myrcvdiffp = 159 peerxmitdiffp = 41091
PPP: myrcvdiffo = 2183 peerxmitdiffo = 1714439
PPP: threshold = 25
PPP Serial4(i): rlqr transmit failure. successes = 15
PPP: myxmitdiffp = 41091 peerrcvdiffp = 159
PPP: myxmitdiffo = 1714439 peerrcvdiffo = 2183
PPP: l->OutLQRs = 1 LastOutLQRs = 1
PPP: threshold = 25
PPP Serial3(i): lqr_protrej() Stop sending LQRs.
PPP Serial3(i): The link appears to be looped back.
```

The following table describes the significant fields shown in the display.

Table 72: debug ppp error Field Descriptions

Field	Description
PPP	PPP debugging output.

Field	Description
Serial3(i)	Interface number associated with this debugging information; indicates that this is an input packet.
rlqr receive failure	Request to negotiate the Quality Protocol option is not accepted.
myrcvdiffp = 159	Number of packets received over the time period.
peerxmitdiffp = 41091	Number of packets sent by the remote node over this period.
myrcvdiffo = 2183	Number of octets received over this period.
peerxmitdiffo = 1714439	Number of octets sent by the remote node over this period.
threshold = 25	Maximum error percentage acceptable on this interface. This percentage is calculated by the threshold value entered in the **ppp quality** *number* interface configuration command. A value of 100 - *number* (100 minus *number*) is the maximum error percentage. In this case, a *number* of 75 was entered. This means that the local router must maintain a minimum 75 percent non-error percentage, or the PPP link will be considered down.
OutLQRs = 1	Local router's current send LQR sequence number.
LastOutLQRs = 1	The last sequence number that the remote node side has seen from the local node.

The following is sample output from the **debug ppp authentication** command. Use this command to determine why an authentication fails.

```
Router# debug ppp authentication
Serial0: Unable to authenticate. No name received from peer
Serial0: Unable to validate CHAP response. USERNAME pioneer not found.
Serial0: Unable to validate CHAP response. No password defined for USERNAME pioneer
Serial0: Failed CHAP authentication with remote.
Remote message is Unknown name
Serial0: remote passed CHAP authentication.
Serial0: Passed CHAP authentication with remote.
Serial0: CHAP input code = 4 id = 3 len = 48
```

In general, these messages are self-explanatory. Fields that can show optional output are outlined in the following table.

Table 73: debug ppp authentication Field Descriptions

Field	Description
Serial0	Interface number associated with this debugging information and CHAP access session in question.
USERNAME pioneer not found.	The name *pioneer* in this example is the name received in the CHAP response. The router looks up this name in the list of usernames that are configured for the router.

Field	Description
Remote message is Unknown name	The following messages can appear: • No name received to authenticate • Unknown name • No secret for given name • Short MD5 response received • MD compare failed
code = 4	Specific CHAP type packet detected. Possible values are as follows: • 1--Challenge • 2--Response • 3--Success • 4--Failure
id = 3	ID number per LCP packet format.
len = 48	Packet length without header.

The following shows sample output from the **debug ppp** command usingthe **cbcp** keyword**.** This output depicts packet exchanges under normal PPP operation where the Cisco access server is waiting for the remote PC to respond to the MSCB request. The router also has **debug ppp negotiation** and **service timestamps msec** commands enabled.

```
Router# debug ppp cbcp
Dec 17 00:48:11.302: As8 MCB: User mscb Callback Number - Client ANY
Dec 17 00:48:11.306: Async8 PPP: O MCB Request(1) id 1 len 9
Dec 17 00:48:11.310: Async8 MCB: O  1  1  0  9  2  5  0  1  0
Dec 17 00:48:11.314: As8 MCB: O Request Id 1 Callback Type Client-Num delay 0
Dec 17 00:48:13.342: As8 MCB: Timeout in state WAIT_RESPONSE
Dec 17 00:48:13.346: Async8 PPP: O MCB Request(1) id 2 len 9
Dec 17 00:48:13.346: Async8 MCB: O  1  2  0  9  2  5  0  1  0
Dec 17 00:48:13.350: As8 MCB: O Request Id 2 Callback Type Client-Num delay 0
Dec 17 00:48:15.370: As8 MCB: Timeout in state WAIT_RESPONSE
Dec 17 00:48:15.374: Async8 PPP: O MCB Request(1) id 3 len 9
Dec 17 00:48:15.374: Async8 MCB: O  1  3  0  9  2  5  0  1  0
Dec 17 00:48:15.378: As8 MCB: O Request Id 3 Callback Type Client-Num delay 0
Dec 17 00:48:17.398: As8 MCB: Timeout in state WAIT_RESPONSE
Dec 17 00:48:17.402: Async8 PPP: O MCB Request(1) id 4 len 9
Dec 17 00:48:17.406: Async8 MCB: O  1  4  0  9  2  5  0  1  0
Dec 17 00:48:17.406: As8 MCB: O Request Id 4 Callback Type Client-Num delay 0
Dec 17 00:48:19.426: As8 MCB: Timeout in state WAIT_RESPONSE
Dec 17 00:48:19.430: Async8 PPP: O MCB Request(1) id 5 len 9
Dec 17 00:48:19.430: Async8 MCB: O  1  5  0  9  2  5  0  1  0
Dec 17 00:48:19.434: As8 MCB: O Request Id 5 Callback Type Client-Num delay 0
Dec 17 00:48:21.454: As8 MCB: Timeout in state WAIT_RESPONSE
Dec 17 00:48:21.458: Async8 PPP: O MCB Request(1) id 6 len 9
Dec 17 00:48:21.462: Async8 MCB: O  1  6  0  9  2  5  0  1  0
Dec 17 00:48:21.462: As8 MCB: O Request Id 6 Callback Type Client-Num delay 0
Dec 17 00:48:23.482: As8 MCB: Timeout in state WAIT_RESPONSE
```

```
Dec 17 00:48:23.486: Async8 PPP: O MCB Request(1) id 7 len 9
Dec 17 00:48:23.490: Async8 MCB: O  1  7  0  9  2  5  0  1  0
Dec 17 00:48:23.490: As8 MCB: O Request Id 7 Callback Type Client-Num delay 0
Dec 17 00:48:25.510: As8 MCB: Timeout in state WAIT_RESPONSE
Dec 17 00:48:25.514: Async8 PPP: O MCB Request(1) id 8 len 9
Dec 17 00:48:25.514: Async8 MCB: O  1  8  0  9  2  5  0  1  0
Dec 17 00:48:25.518: As8 MCB: O Request Id 8 Callback Type Client-Num delay 0
Dec 17 00:48:26.242: As8 PPP: I pkt type 0xC029, datagramsize 18
Dec 17 00:48:26.246: Async8 PPP: I MCB Response(2) id 8 len 16
Dec 17 00:48:26.250: Async8 MCB: I  2  8  0  10  2  C  C  1  32  34  39  32  36  31  33  0

Dec 17 00:48:26.254: As8 MCB: Received response
Dec 17 00:48:26.258: As8 MCB: Response CBK-Client-Num 2 12 12,  addr 1-2492613
Dec 17 00:48:26.262: Async8 PPP: O MCB Ack(3) id 9 len 16
Dec 17 00:48:26.266: Async8 MCB: O  3  9  0  10  2  C  C  1  32  34  39  32  36  31  33  0

Dec 17 00:48:26.270: As8 MCB: O Ack Id 9 Callback Type Client-Num delay 12
Dec 17 00:48:26.270: As8 MCB: Negotiated MCB with peer
Dec 17 00:48:26.390: As8 LCP: I TERMREQ [Open] id 4 len 8 (0x00000000)
Dec 17 00:48:26.390: As8 LCP: O TERMACK [Open] id 4 len 4
Dec 17 00:48:26.394: As8 MCB: Peer terminating the link
Dec 17 00:48:26.402: As8 MCB: Initiate Callback for mscb at 2492613 using Async
```

The following is sample output from the **debug ppp compression** command with **service timestamps** enabled and shows a typical PPP packet exchange between the router and Microsoft client where the MPPC header sequence numbers increment correctly:

```
Router# debug ppp compression
00:04:14: BR0:1 MPPC: Decomp - hdr/exp_cc# 0x2003/0x0003
00:04:14: BR0:1 MPPC: Decomp - hdr/exp_cc# 0x2004/0x0004
00:04:14: BR0:1 MPPC: Decomp - hdr/exp_cc# 0x2005/0x0005
00:04:14: BR0:1 MPPC: Decomp - hdr/exp_cc# 0x2006/0x0006
00:04:14: BR0:1 MPPC: Decomp - hdr/exp_cc# 0x2007/0x0007
```

The following table describes the significant fields shown in the display.

Table 74: debug ppp compression Field Descriptions

Field	Description
interface	Interface enabled with MPPC.
Decomp - hdr/	Decompression header and bit settings.
exp_cc#	Expected coherency count.
0x2003	Received sequence number.
0x0003	Expected sequence number.

The following shows sample output from **debug ppp negotiation** and **debug ppp error** commands, which can be used to troubleshoot initial PPP negotiation and setup errors. This example shows a virtual interface (virtual interface 1) during normal PPP operation and CCP negotiation.

```
Router# debug ppp negotiation error
Vt1 PPP: Unsupported or un-negotiated protocol. Link arp
VPDN: Chap authentication succeeded for p5200
Vi1 PPP: Phase is DOWN, Setup
Vi1 VPDN: Virtual interface created for dinesh@cisco.com
Vi1 VPDN: Set to Async interface
```

```
Vi1 PPP: Phase is DOWN, Setup
Vi1 VPDN: Clone from Vtemplate 1 filterPPP=0 blocking
Vi1 CCP: Re-Syncing history using legacy method
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
Vi1 PPP: Treating connection as a dedicated line
Vi1 PPP: Phase is ESTABLISHING, Active Open
Vi1 LCP: O CONFREQ [Closed] id 1 len 25
Vi1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Vi1 LCP:    MagicNumber 0x000FB69F (0x0506000FB69F)
Vi1 LCP:    PFC (0x0702)
Vi1 LCP:    ACFC (0x0802)
Vi1 VPDN: Bind interface direction=2
Vi1 PPP: Treating connection as a dedicated line
Vi1 LCP: I FORCED CONFREQ len 21
Vi1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Vi1 LCP:    MagicNumber 0x12A5E4B5 (0x050612A5E4B5)
Vi1 LCP:    PFC (0x0702)
Vi1 LCP:    ACFC (0x0802)
Vi1 VPDN: PPP LCP accepted sent & rcv CONFACK
Vi1 PPP: Phase is AUTHENTICATING, by this end
Vi1 CHAP: O CHALLENGE id 1 len 27 from "l_4000"
Vi1 CHAP: I RESPONSE id 20 len 37 from "dinesh@cisco.com"
Vi1 CHAP: O SUCCESS id 20 len 4
Vi1 PPP: Phase is UP
Vi1 IPCP: O CONFREQ [Closed] id 1 len 10
Vi1 IPCP:    Address 15.2.2.3 (0x03060F020203)
Vi1 CCP: O CONFREQ [Not negotiated] id 1 len 10
Vi1 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
Vi1 IPCP: I CONFREQ [REQsent] id 1 len 34
Vi1 IPCP:    Address 0.0.0.0 (0x030600000000)
Vi1 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Vi1 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Vi1 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Vi1 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Vi1 IPCP: Using the default pool
Vi1 IPCP: Pool returned 11.2.2.5
Vi1 IPCP: O CONFREJ [REQsent] id 1 len 16
Vi1 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Vi1 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Vi1 CCP: I CONFREQ [REQsent] id 1 len 15
Vi1 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
Vi1 CCP:    Stacker history 1 check mode EXTENDED (0x1105000104)
Vi1 CCP: Already accepted another CCP option, rejecting this STACKER
Vi1 CCP: O CONFREJ [REQsent] id 1 len 9
Vi1 CCP:    Stacker history 1 check mode EXTENDED (0x1105000104)
Vi1 IPCP: I CONFACK [REQsent] id 1 len 10
Vi1 IPCP:    Address 15.2.2.3 (0x03060F020203)
Vi1 CCP: I CONFACK [REQsent] id 1 len 10
Vi1 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
Vi1 CCP: I CONFREQ [ACKrcvd] id 2 len 10
Vi1 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
Vi1 CCP: O CONFACK [ACKrcvd] id 2 len 10
Vi1 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
Vi1 CCP: State is Open
Vi1 IPCP: I CONFREQ [ACKrcvd] id 2 len 22
Vi1 IPCP:    Address 0.0.0.0 (0x030600000000)
Vi1 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Vi1 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Vi1 IPCP: O CONFNAK [ACKrcvd] id 2 len 22
Vi1 IPCP:    Address 11.2.2.5 (0x03060B020205)
Vi1 IPCP:    PrimaryDNS 171.69.1.148 (0x8106AB450194)
Vi1 IPCP:    SecondaryDNS 171.69.2.132 (0x8306AB450284)
```

```
Vi1 IPCP: I CONFREQ [ACKrcvd] id 3 len 22
Vi1 IPCP:    Address 11.2.2.5 (0x03060B020205)
Vi1 IPCP:    PrimaryDNS 171.69.1.148 (0x8106AB450194)
Vi1 IPCP:    SecondaryDNS 171.69.2.132 (0x8306AB450284)
Vi1 IPCP: O CONFACK [ACKrcvd] id 3 len 22
Vi1 IPCP:    Address 11.2.2.5 (0x03060B020205)
Vi1 IPCP:    PrimaryDNS 171.69.1.148 (0x8106AB450194)
Vi1 IPCP:    SecondaryDNS 171.69.2.132 (0x8306AB450284)
Vi1 IPCP: State is Open
Vi1 IPCP: Install route to 11.2.2.5
```

debug ppp bap

To display general Bandwidth Allocation Control Protocol (BACP) transactions, use the **debugpppbap** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ppp bap [{**error** | **event** | **negotiation**}]
no debug ppp bap [{**error** | **event** | **negotiation**}]

Syntax Description

error	(Optional) Displays local errors.
event	(Optional) Displays information about protocol actions and transitions between action states (pending, waiting, idle) on the link.
negotiation	(Optional) Displays successive steps in negotiations between peers.

Command Modes

Privileged EXEC

Usage Guidelines

Do not use this command when memory is scarce or in very high traffic situations.

Examples

The following types of events generate the debugging messages displayed in the figures in this section:

- A dial attempt failed.
- A BACP group was created.
- A BACP group was removed.
- The precedence of the group changed.
- Attempting to dial a number.
- Received a BACP message.
- Discarding a BACP message.
- Received an unknown code.
- Cannot find the appropriate BACP group on input.
- Displaying the response type.
- Incomplete mandatory options notification.
- Invalid outgoing message type.
- Unable to build an output message.
- Sending a BACP message.
- Details about the sent message (type of message, its identifier, the virtual access interface that sent it).

The following is sample output from the **debugpppbap** command:

```
Router# debug ppp bap
BAP Virtual-Access1: group "laudrup" (2) (multilink) without precedence created
BAP laudrup: sending CallReq, id 2, len 38 on BRI3:1 to remote
BAP Virtual-Access1: received CallRsp, id 2, len 13
BAP laudrup: CallRsp, id 2, ACK
BAP laudrup: attempt1 to dial 19995776677 on BRI3
 ---> reason BAP - Multilink bundle overloaded
BAP laudrup: sending StatusInd, id 2, len 44 on Virtual-Access1 to remote
BAP Virtual-Access1: received StatusRsp, id 2, len 1
BAP laudrup: StatusRsp, id 2, ACK
```

The following table describes the significant fields shown in the display.

Table 75: debug ppp bap Field Descriptions

Field	Description
BAP Virtual-Access1:	Identifier of the virtual access interface in use.
group “laudrup”	Name of the BACP group.
sending CallReq	Action initiated; in this case, sending a call request.
on BRI3:1 to remote	Physical interface being used.
BAP laudrup: attempt1 to dial 19995776677 on BRI3 ---> reason BAP - Multilink bundle overloaded	Call initiated, number being dialed, and physical interface being used. Reason for initiating the BACP call.
BAP laudrup: sending StatusInd, id 2, len 44 on Virtual-Access1 to remote	Details about the sent message: It was a status indication message, had identifier 2, had a BACP datagram length 44, and was sent on virtual access interface 1. You can display information about the virtual access interface by using the **showinterfacesvirtual-access** EXEC command. (The length shown at the end of each negotiated option includes the 2-byte type and length header.)

The **debugpppbapevent** command might show state transitions and protocol actions, in addition to the basic **debugpppbap** command.

The following is sample output from the **debugpppbapevent** command:

```
Router# debug ppp bap event
BAP laudrup: Idle --> AddWait
BAP laudrup: AddWait --> AddPending
BAP laudrup: AddPending --> Idle
```

The following is sample output from the **debugpppbapevent** command:

```
Router# debug ppp bap event
Peer does not support a message type
No response to a particular request
No response to all request retransmissions
Not configured to initiate link addition
Expected action by peer has not occurred
```

```
Exceeded number of retries
No links available to call out
Unable to provide phone numbers for callback
Maximum number of links in the group
Minimum number of links in the group
Unable to process link addition at present
Unable to process link removal at present
Not configured/unable to initiate link removal
Link addition completed notification
Link addition failed notification
Determination of location of the group config
Link with specified discriminator not in group
Link removal failed
Call failure with status
Failed to dial specified number
Discarding retransmission
Unable to find received identifier
Received StatusInd when no call pending
Discarding message with no phone delta
Unable to send message in particular state
Received a zero identifier
Request has precedence
```

The error messages displayed might be added to the basic output when the **debugpppbaperror** command is used. Because the errors are very rare, you might never see these messages.

```
Router# debug ppp bap error
Unable to find appropriate request for received response
Invalid message type of queue
Received request is not part of the group
Add link attempt failed to locate group
Remove link attempt failed to locate group
Unable to inform peer of link addition
Changing of precedence cannot locate group
Received short header/illegal length/short packet
Invalid configuration information length
Unable to NAK incomplete options
Unable to determine current number of links
No interface list to dial on
Attempt to send invalid data
Local link discriminator is not in group
Received response type is incorrect for identifier
```

The messages displayed might be added to the basic output when the **debugpppbapnegotiation** command is used:

```
Router# debug ppp bap negotiation
BAP laudrup: adding link speed 64 kbps for type 0x1 len 5
BAP laudrup: adding reason "User initiated addition", len 25
BAP laudrup: CallRsp, id 4, ACK
BAP laudrup: link speed 64 kbps for types 0x1, len 5 (ACK)
BAP laudrup: phone number "1: 0 2: ", len 7 (ACK)
BAP laudrup: adding call status 0, action 0 len 4
BAP laudrup: adding 1 phone numbers "1: 0 2: " len 7
BAP laudrup: adding reason "Successfully added link", len 25
BAP laudrup: StatusRsp, id 4, ACK
```

Additional negotiation messages might also be displayed for the following:

```
Received BAP message
Sending message
```

```
Decode individual options for send/receive
Notification of invalid options
```

The following shows additional reasons for a particular BAP action that might be displayed in an "adding reason" line of the **debugpppbapnegotiation** command output:

```
"Outgoing add request has precedence"
"Outgoing remove request has precedence"
"Unable to change request precedence"
"Unable to determine valid phone delta"
"Attempting to add link"
"Link addition is pending"
"Attempting to remove link"
"Link removal is pending"
"Precedence of peer marked CallReq for no action"
"Callback request rejected due to configuration"
"Call request rejected due to configuration"
"No links of specified type(s) available"
"Drop request disallowed due to configuration"
"Discriminator is invalid"
"No response to call requests"
"Successfully added link"
"Attempt to dial destination failed"
"No interfaces present to dial out"
"No dial string present to dial out"
"Mandatory options incomplete"
"Load has not exceeded threshold"
"Load is above threshold"
"Currently attempting to dial destination"
"No response to CallReq from race condition"
```

The following table describes the reasons for a BACP Negotiation Action.

Table 76: Explanation of Reasons for BACP Negotiation Action

Reason	Explanation
"Outgoing add request has precedence"	Received a CallRequest or CallbackRequest while we were waiting on a CallResponse or CallbackResponse to a sent request. We are the favored peer from the initial BACP negotiation, so we are issuing a NAK to our peer request.
"Outgoing remove request has precedence"	Received a LinkDropQueryRequest while waiting on a LinkDropQueryResponse to a sent request. We are the favored peer from the initial BACP negotiation, therefore we are issuing a NAK to our peer request.
"Unable to change request precedence"	Received a CallRequest, CallbackRequest, or LinkDropQueryRequest while waiting on a LinkDropQueryResponse to a sent request. Our peer is deemed to be the favored peer from the initial BACP negotiation and we were unable to change the status of our outgoing request in response to the favored request, so we are issuing a NAK. (This is an internal error and should never be seen.)
"Unable to determine valid phone delta"	Received a CallRequest from our peer but are unable to provide the required phone delta for the response, so we are issuing a NAK. (This is an internal error and should never be seen.)

Reason	Explanation
“Attempting to add link”	Received a LinkDropQueryRequest while attempting to add a link; a NAK is issued.
“Link addition is pending”	Received a LinkDropQueryRequest, CallRequest, or CallbackRequest while attempting to add a link as the result of a previous operation; a NAK is issued in the response.
“Attempting to remove link”	Received a CallRequest or CallbackRequest while attempting to remove a link; a NAK is issued.
“Link removal is pending”	Received a CallRequest, CallbackRequest, or LinkDropQueryRequest while attempting to remove a link as the result of a previous operation; a NAK is issued in the response.
“Precedence of peer marked CallReq for no action”	Received an ACK to a previously unfavored CallRequest; we are issuing a CallStatusIndication to inform our peer that there will be no further action on our part as per this response.
“Callback request rejected due to configuration”	Received a CallbackRequest but we are configured not to accept them; a REJect is issued to our peer.
“Call request rejected due to configuration”	Received a CallRequest but we are configured not to accept them; a REJect is issued to our peer.
“No links of specified type(s) available”	We received a CallRequest but no links of the specified type and speed are available; a NAK is issued.
“Drop request disallowed due to configuration”	Received a LinkDropQueryRequest but we are configured not to accept them; a NAK is issued to our peer.
“Discriminator is invalid”	Received a LinkDropQueryRequest but the local link discriminator is not contained within the bundle; a NAK is issued.
“No response to call requests”	After no response to our CallRequest message, a CallStatusIndication is sent to the peer informing that no more action will be taken on behalf of this operation.
“Successfully added link”	Sent as part of the CallStatusIndication informing our peer that we successfully completed the addition of a link to the bundle as the result of the transmission of a CallRequest or the reception of a CallbackRequest.
“Attempt to dial destination failed”	Sent as part of the CallStatusIndication informing our peer that we failed in an attempt to add a link to the bundle as the result of the transmission of a CallRequest or the reception of a CallbackRequest. The retry field with the CallStatusIndication informs the peer of our intentions.
“No interfaces present to dial out”	There are no available interfaces to dial out on to attempt to add a link to the bundle, and we will not retry the dial attempt.
“No dial string present to dial out”	We do not have a dial string to dial out with to attempt to add a link to the bundle, and we are not going to retry the dial attempt. (This is an internal error and should never be seen.)

Reason	Explanation
“Mandatory options incomplete”	Received a CallRequest, CallbackRequest, LinkDropQueryRequest, or CallStatusIndication and the mandatory options are not present, so a NAK is issued in the response. (A CallStatusResponse is an ACK, however).
“Load has not exceeded threshold”	Received a CallRequest or CallbackRequest but we are issuing a NAK in the response. We are monitoring the load of the bundle, and so we determine when links should be added to the bundle.
“Load is above threshold”	Received a LinkDropQueryRequest but we are issuing a NAK in the response. We are monitoring the load of the bundle, and so we determine when links should be removed from the bundle.
“Currently attempting to dial destination”	Received a CallbackRequest which is a retransmission of one that we previously ACK’d and are dialing the number suggested in the request. We are issuing an ACK because we did so previously, even though our peer never saw the previous response.
“No response to CallReq from race condition”	We issued a CallRequest but failed to receive a response, and we are issuing a CallStatusIndication to inform our peer of our intention not to proceed with the operation.

debug ppp ip address-save

To display debug information about the IPv4 Address Conservation in Dual Stack Environments feature such as authorization, authentication, and IPv4 address allocation messages on the broadband remote access server (BRAS), use the **debug ppp ip address-save** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ppp ip address-save
no debug ppp ip address-save

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.5S	This command was introduced.

Usage Guidelines

Use the **debug ppp ip address-save** command to display authorization, authentication, and IPv4 address allocation messages on the BRAS. This command shows that the IPv4 Address Conservation in Dual Stack Environments feature has been enabled and displays the events that are triggered by enabling the feature. See the "Related Commands" section for **debug** commands that should be used in conjunction with this command

Examples

The following is sample output from the **debug ppp ip address-save** command:

```
Router# debug ppp ip address-save

Vi2.1 IPCP AUTH: Adding password in AAA author request
Vi2.1 IPCP AUTH: Added password and AAA VSA [enable] in author request
Vi2.1 PPP: Added IPv4 address [10.1.1.25] to include in acct record
Vi2.1 PPP: Triggering interim acct request
Vi2.1 PPP: IPCP going down, resetting neg authorized flag
Vi2.1 PPP: Peer IPv4 address in author data = 10.1.1.25
Vi2.1 PPP: Removing IPv4 address from Accounting DB
Vi2.1 PPP: Triggering interim acct request
Vi2.1 PPP: IPCP went down, checking status of other NCPs
```

The output is self-explanatory.

Related Commands

Command	Description
debug ppp authentication	Displays authentication protocol messages, including CHAP packet exchanges and PAP exchanges.
debug ppp authorization	Displays information about authorization attributes received from the RADIUS server.
debug ppp negotiation	Displays PPP packets sent during PPP startup, where PPP options are negotiated.

Command	Description
debug radius	Displays accounting and authentication information and client/server interaction events on the RADIUS server.
ppp ip address-save aaa-acct-vsa	Enables IPv4 address conservation on the BRAS.

debug ppp multilink events

To display information about events affecting multilink groups established for Bandwidth Allocation Control Protocol (BACP), use the**debugpppmultilinkevents** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ppp multilink events
no debug ppp multilink events

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Caution Do not use this command when memory is scarce or in very high traffic situations.

Examples

The following is sample output from the **debugpppmultilinkevents** command:

```
Router# debug ppp multilink events
MLP laudrup: established BAP group 4 on Virtual-Access1, physical BRI3:1
MLP laudrup: removed BAP group 4
```

Other event messages include the following:

```
Unable to find bundle for BAP group identifier
Unable to find physical interface to start BAP
Unable to create BAP group
Attempt to start BACP when inactive or running
Attempt to start BACP on non-MLP interface
Link protocol has gone down, removing BAP group
Link protocol has gone down, BAP not running or present
```

The following table describes the significant fields shown in the display.

Table 77: debug ppp multilink events Field Descriptions

Field	Description
MLP laudrup	Name of the multilink group.
established BAP group 4	Internal identifier. The same identifiers are used in the **showpppbapgroup** command output.
Virtual-Access1	Dynamic access interface number.
physical BRI3:1	Bundle was established from a call on this interface.
removed BAP group 4	When the bundle is removed, the associated BACP group (with its ID) is also removed.

debug ppp multilink fragments

To display information about individual multilink fragments and important multilink events, use the **debug ppp multilink fragments** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ppp multilink fragments
no debug ppp multilink fragments

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Caution The **debug ppp multilink fragments** command has some memory overhead and should not be used when memory is scarce or in very high traffic situations.

Examples

The following is sample output from the **debug ppp multilink fragments** command when used with the **ping** EXEC command. The debug output indicates that a multilink PPP packet on interface BRI 0 (on the B channel) is an input (I) or output (O) packet. The output also identifies the sequence number of the packet and the size of the fragment.

```
Router# debug ppp multilink fragments

Router# ping 7.1.1.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.1.1.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
Router#
2:00:28: MLP BRI0: B-Channel 1: O seq 80000000: size 58
2:00:28: MLP BRI0: B-Channel 2: O seq 40000001: size 59
2:00:28: MLP BRI0: B-Channel 2: I seq 40000001: size 59
2:00:28: MLP BRI0: B-Channel 1: I seq 80000000: size 58
2:00:28: MLP BRI0: B-Channel 1: O seq 80000002: size 58
2:00:28: MLP BRI0: B-Channel 2: O seq 40000003: size 59
2:00:28: MLP BRI0: B-Channel 2: I seq 40000003: size 59
2:00:28: MLP BRI0: B-Channel 1: I seq 80000002: size 58
2:00:28: MLP BRI0: B-Channel 1: O seq 80000004: size 58
2:00:28: MLP BRI0: B-Channel 2: O seq 40000005: size 59
2:00:28: MLP BRI0: B-Channel 2: I seq 40000005: size 59
2:00:28: MLP BRI0: B-Channel 1: I seq 80000004: size 58
2:00:28: MLP BRI0: B-Channel 1: O seq 80000006: size 58
2:00:28: MLP BRI0: B-Channel 2: O seq 40000007: size 59
2:00:28: MLP BRI0: B-Channel 2: I seq 40000007: size 59
2:00:28: MLP BRI0: B-Channel 1: I seq 80000006: size 58
2:00:28: MLP BRI0: B-Channel 1: O seq 80000008: size 58
2:00:28: MLP BRI0: B-Channel 2: O seq 40000009: size 59
2:00:28: MLP BRI0: B-Channel 2: I seq 40000009: size 59
2:00:28: MLP BRI0: B-Channel 1: I seq 80000008: size 58
```

debug ppp multilink negotiation

Note Effective with release 11.3, the **debugpppmultilinknegotiation** command is not available in Cisco IOS software.

To display information about events affecting multilink groups established controlled by Bandwidth Allocation Control Protocol (BACP), use the**debugpppmultilinknegotiation** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ppp multilink negotiation
no debug ppp multilink negotiation

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3	This command was removed and is not available in Cisco IOS software.

Usage Guidelines

Caution Do not use this command when memory is scarce or in very high traffic situations.

Examples

The following sample output shows Link Control Protocol (LCP) and Network Control Program (NCP) messages that might appear in **debugpppmultilinknegotiation** command. These messages show information about PPP negotiations between the multilink peers.

```
Router# debug ppp multilink negotiation
ppp: sending CONFREQ, type = 23 (CI_LINK_DISCRIMINATOR), value = 0xF
PPP BRI3:1: received config for type = 23 (LINK_DISCRIMINATOR) value = 0xA acked
Router# debug ppp multilink negotiation
ppp: sending CONFREQ, type = 1 (CI_FAVORED_PEER), value = 0x647BD090
PPP Virtual-Access1: received CONFREQ, type 1, value = 0x382BBF5 (ACK)
PPP Virtual-Access1: BACP returning CONFACK
ppp: config ACK received, type = 1 (CI_FAVORED_PEER), value = 0x647BD090
PPP Virtual-Access1: BACP up
```

The following table describes the significant fields shown in the display.

Table 78: debug ppp multilink negotiation Field Descriptions

Field	Description
sending CONFREQ, type = 23 (CI_LINK_DISCRIMINATOR), value = 0xF	Sending a configuration request and the value of the link discriminator. Each peer assigns a discriminator value to identify a specific link. The values are significant to each peer individually but do not have to be shared.
PPP BRI3:1:	Physical interface being used.
CI_FAVORED_PEER	When the PPP NCP negotiation occurs over the first link in a bundle, the BACP peers use a Magic Number akin to that used by LCP to determine which peer should be favored when both implementations send a request at the same time. The peer that negotiated the higher number is deemed to be favored. That peer should issue a negative acknowledgment to its unfavored peer, which in turn should issue a positive acknowledgment, if applicable according to other link considerations.
PPP Virtual-Access1: BACP returning CONFACK	Returning acknowledgment that BACP is configured.
PPP Virtual-Access1: BACP up	Indicating that the BACP NCP is open.

debug ppp redundancy

To debug PPP synchronization on the networking device, use the **debug ppp redundancy**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug ppp redundancy [{**detailed** | **event**}]
no debug ppp redundancy [{**detailed** | **event**}]

Syntax Description

detailed	(Optional) Displays detailed debug messages related to specified PPP redundancy events.
event	(Optional) Displays information about protocol actions and transitions between action states (pending, waiting, idle) on the link.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(22)S	This command was introduced on the Cisco 7500, 10000, and 12000 series Internet routers.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S on Cisco 7500 series routers.
12.2(20)S	Support was added for the Cisco 7304 router. The Cisco 7500 series router is not supported in Cisco IOS Release 12.2(20)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

The following example displays detailed debug messages related to specified PPP redundancy events:

```
Router# debug ppp redundancy detailed
```

debug ppp unique address

To display debugging information about duplicate addresses received from RADIUS, use the **debug ppp unique address**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ipv6 policy
no debug ipv6 policy

Syntax Description

This command has no arguments or keywords.

Command Default

Information about duplicate addresses received from RADIUS is not displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced.

Usage Guidelines

The **debug p**pp unique address command enables you to view debugging information about duplicate addresses received from RADIUS.

Examples

The following example enables debugging output about duplicate addresses received from RADIUS:

```
Router# debug ppp unique address
```

debug pppatm

To enable debug reports for PPP over ATM (PPPoA) events, errors, and states, either globally or conditionally, on an interface or virtual circuit (VC), use the **debugpppatm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug pppatm {**event** | **error** | **state**} [**interface atm** *interface-number* [*subinterface-number*]] **vc** {[*vpi*/*vci*] *vcivirtual-circuit-name*}
no debug pppatm {**event** | **error** | **state**} [**interface atm** *interface-number* [*subinterface-number*]] **vc** {[*vpi*/*vci*] *vcivirtual-circuit-name*}

Syntax Description

event	PPPoA events.
error	PPPoA errors.
state	PPPoA state.
interface atm *interface-number* [*subinterface-number*]	(Optional) Specifies a particular ATM interface by interface number and optionally a subinterface number separated by a period.
vc [*vpi*/]*vcivirtual-circuit-name*	(Optional) Virtual circuit (VC) keyword followed by a virtual path identifier (VPI), virtual channel identifier (VCI), and VC name. A slash mark is required after the VPI.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Usage Guidelines

Each specific PPPoA debug report must be requested on a separate command line; see the "Examples" section.

Examples

The following is example output of a PPPoA session with event, error, and state debug reports enabled on ATM interface 1/0.10:

```
Router# debug pppatm event interface atm1/0.10
Router# debug pppatm error interface atm1/0.10
Router# debug pppatm state interface atm1/0.10
00:03:08: PPPATM: ATM1/0.10 0/101 [1], Event = Clear Session
00:03:08: PPPATM: ATM1/0.10 0/101 [1], Event = Disconnecting
00:03:08: PPPATM: ATM1/0.10 0/101 [1], Event = AAA gets dynamic attrs
00:03:08: PPPATM: ATM1/0.10 0/101 [1], Event = AAA gets dynamic attrs
00:03:08: PPPATM: ATM1/0.10 0/101 [1], Event = SSS Cleanup
00:03:08: PPPATM: ATM1/0.10 0/101 [0], State = DOWN
00:03:08: PPPATM: ATM1/0.10 0/101 [0], Event = Up Pending
00:03:16: PPPATM: ATM1/0.10 0/101 [0], Event = Up Dequeued
00:03:16: PPPATM: ATM1/0.10 0/101 [0], Event = Processing Up
```

```
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = Access IE allocated
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = Set Pkts to SSS
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = AAA gets retrived attrs
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = AAA gets nas port details
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = AAA gets dynamic attrs
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = AAA gets dynamic attrs
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = AAA unique id allocated
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = No AAA method list set
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = SSS Request
00:03:16: PPPATM: ATM1/0.10 0/101 [2], State = NAS_PORT_POLICY_INQUIRY
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = SSS Msg Received = 1
00:03:16: PPPATM: ATM1/0.10 0/101 [2], State = PPP_START
00:03:16: PPPATM: ATM1/0.10 0/101 [2], Event = PPP Msg Received = 1
00:03:16: PPPATM: ATM1/0.10 0/101 [2], State = LCP_NEGOTIATION
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = PPP Msg Received = 4
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = HW Switch support FORW = 0
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = Access IE get nas port
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = AAA gets dynamic attrs
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = AAA gets dynamic attrs
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = PPP Msg Received = 5
00:03:27: PPPATM: ATM1/0.10 0/101 [2], Event = Set Pkts to SSS
00:03:27: PPPATM: ATM1/0.10 0/101 [2], State = FORWARDED
```

The following table describes the significant fields shown in the display.

Table 79: debug pppatm Field Descriptions

Field	Description
Event	Reports PPPoA events for use by Cisco engineering technical assistance personnel.
State	Reports PPPoA states for use by Cisco engineering technical assistance personnel.

Related Commands

Command	Description
atm pppatm passive	Places an ATM subinterface into passive mode.
show pppatm summary	Displays PPPoA session counts.

debug pppatm redundancy

To debug PPP over ATM (PPPoA) redundancy events on a dual Route Processor High Availability (HA) system and display cluster control manager (CCM) events and messages, use the **debug pppatm redundancy**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug pppatm redundancy [**interface atm** *interface-number* [**vc** {*vpi*/*vcivci*}]]
no debug pppatm redundancy [**interface atm** *interface-number* [**vc** {*vpi*/*vcivci*}]]

Syntax Description

interface atm *interface-number*	(Optional) Specifies a particular ATM interface by interface number.
vc	(Optional) Specifies the virtual circuit (VC).
vpi/vci	(Optional) Virtual path identifier (VPI) and virtual channel identifier (VCI) value. The range is from 0 to 255.
vci	(Optional) VCI. The range is from 1 to 65535.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(31)SB2	This command was introduced.
Cisco IOS XE Release 3.3S	This command was modified. The **interface atm** *interface-number*keyword-argument pair, **vc**keyword, and *vpi/vci*and *vci*arguments were added.

Usage Guidelines

The CCM provides the capability to facilitate and synchronize session bring-up on the standby processor of a dual Route Processor HA system. Use the **debug pppatm redundancy** command to display CCM events and messages for PPPoA sessions on HA systems.

To create sessions on the standby processor with the same virtual-access (sub)interface as that on the active processor, base virtual-access interface creation on the standby processor is delayed until the first PPPoA session synchronizes to the standby processor. For each session, PPPoA synchronizes information elements such as virtual access (VAccess) descriptor, physical software for interface descriptor block (swidb) descriptor, switch handle, segment handle, and ATM virtual circuit's (VC) virtual path identifier (VPI) and virtual channel identifier (VCI) numbers to the standby processor. The **interface atm**keywords and *interface-number* argument specify a particular ATM interface by interface number and the **vc**keyword specifies the VC.

Note The debug pppatm redundancy command does not display output on the active processor during normal synchronization; that is, the command displays output on the active processor only during an error condition.

Note This command is used only by Cisco engineers for internal debugging of CCM processes.

Examples

The following is sample output from the debug pppatm redundancy command from a Cisco 10000 series router active processor, along with sample output from the **show pppatm redundancy** command from the standby processor. No field descriptions are provided because command output is used for Cisco internal debugging purposes only.

```
Router# debug pppatm redundancy
PPP over ATM redundancy debugging is on
Router-stby# show pppatm redundancy
0 : Session recreate requests from CCM
0 : Session up events invoked
0 : Sessions reaching PTA
0 : Sessions closed by CCM
0 : Session down events invoked
0 : Queued sessions waiting for base hwidb creation
0 : Sessions queued for VC up notification so far
0 : Sessions queued for VC encap change notification so far
0 : VC activation notifications received from ATM
0 : VC encap change notifications received from ATM
0 : Total queued sessions waiting for VC notification(Encap change+VC Activation)
```

Related Commands

Command	Description
debug pppatm	Enables debug reports for PPPoA events, errors, and states, either globally or conditionally, on an interface or VC.

debug pppoe

To display debugging information for PPP over Ethernet (PPPoE) sessions, use the **debugpppoe**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug pppoe {{**data** | **errors** | **events** | **packets**} [{**rmac** *remote-mac-address* | **interface** *type number* [**vc** {[*vpi*/] *vcivc-name*}] [**vlan** *vlan-id*]}] | **elog**}
no debug pppoe {{**data** | **errors** | **events** | **packets**} [{**rmac** *remote-mac-address* | **interface** *type number* [**vc** {[*vpi*/] *vcivc-name*}] [**vlan** *vlan-id*]}] | **elog**}

Syntax Description

data	Displays data packets of PPPoE sessions.
errors	Displays PPPoE protocol errors that prevent a session from being established, or displays errors that cause an established session to be closed.
events	Displays PPPoE protocol messages about events that are part of normal session establishment or shutdown.
packets	Displays each PPPoE protocol packet that is exchanged.
rmac *remote-mac-address*	(Optional) Remote MAC address. Debugging information for PPPoE sessions sourced from this address will be displayed.
interface *type number*	(Optional) Interface for which PPPoE session debugging information will be displayed.
vc	(Optional) Displays debugging information for PPPoE sessions for a specific permanent virtual circuit (PVC).
vpi /	(Optional) ATM network virtual path identifier (VPI) for the PVC. The *vpi* value defaults to 0.
vci	(Optional) ATM network virtual channel identifier (VCI) for the PVC.
vc-name	(Optional) Name of the PVC.
vlan *vlan-id*	(Optional) IEEE 802.1Q VLAN identifier.
elog	Displays PPPoE error logs.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(13)T	This command was introduced. This command replaces the **debugvpdnpppoe-data**, **debugvpdnpppoe-error,debugvpdnpppoe-events,**and**debugvpdnpppoe-packet**commands available in previous Cisco IOS releases.
12.2(15)T	This command was modified to display debugging information on a per-MAC address, per-interface, and per-VC basis.

Release	Modification
12.3(2)T	The **vlan***vlan-id* keyword and argument were added.
12.3(7)XI3	This command was integrated into Cisco IOS Release 12.3(7)XI3.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.
Cisco IOS XE Release 2.1	This command was implemented on Cisco ASR 1000 series routers.

Examples

The following examples show sample output from the**debugpppoe** command:

```
Router# debug pppoe events interface atm 1/0.10 vc 101

PPPoE protocol events debugging is on
Router#
00:41:55:PPPoE 0:I PADI  R:00b0.c2e9.c470 L:ffff.ffff.ffff 0/101 ATM1/0.10
00:41:55:PPPoE 0:O PADO, R:00b0.c2e9.c470 L:0001.c9f0.0c1c 0/101 ATM1/0.10
00:41:55:PPPoE 0:I PADR  R:00b0.c2e9.c470 L:0001.c9f0.0c1c 0/101 ATM1/0.10
00:41:55:PPPoE :encap string prepared
00:41:55:[3]PPPoE 3:Access IE handle allocated
00:41:55:[3]PPPoE 3:pppoe SSS switch updated
00:41:55:[3]PPPoE 3:AAA unique ID allocated
00:41:55:[3]PPPoE 3:No AAA accounting method list
00:41:55:[3]PPPoE 3:Service request sent to SSS
00:41:55:[3]PPPoE 3:Created  R:0001.c9f0.0c1c L:00b0.c2e9.c470 0/101 ATM1/0.10
00:41:55:[3]PPPoE 3:State REQ_NASPORT    Event MORE_KEYS
00:41:55:[3]PPPoE 3:O PADS  R:00b0.c2e9.c470 L:0001.c9f0.0c1c 0/101 ATM1/0.10
00:41:55:[3]PPPoE 3:State START_PPP    Event DYN_BIND
00:41:55:[3]PPPoE 3:data path set to PPP
00:41:57:[3]PPPoE 3:State LCP_NEGO    Event PPP_LOCAL
00:41:57:PPPoE 3/SB:Sent vtemplate request on base Vi2
00:41:57:[3]PPPoE 3:State CREATE_VA    Event VA_RESP
00:41:57:[3]PPPoE 3:Vi2.1 interface obtained
00:41:57:[3]PPPoE 3:State PTA_BIND    Event STAT_BIND
00:41:57:[3]PPPoE 3:data path set to Virtual Acess
00:41:57:[3]PPPoE 3:Connected PTA
Router# debug pppoe errors interface atm 1/0.10
PPPoE protocol errors debugging is on
Router#
00:44:30:PPPoE 0:Max session count(1) on mac(00b0.c2e9.c470) reached.
00:44:30:PPPoE 0:Over limit or Resource low. R:00b0.c2e9.c470 L:ffff.ffff.ffff 0/101 ATM1/0.10
```

The following table describes the significant fields shown in the displays.

Table 80: debug pppoe Field Descriptions

Field	Description
PPPoE	PPPoE debug message header.
0:	PPPoE session ID.

Field	Description
I PADI	Incoming PPPoE Active Discovery Initiation packet.
R:	Remote MAC address.
L:	Local MAC address.
0/101	VPI VCI of the PVC.
ATM1/0.10	Interface type and number.
O PADO	Outgoing PPPoE Active Discovery Offer packet.
I PADR	Incoming PPPoE Active Discovery Request packet.
[3]	Unique user session ID. The same ID is used for identifying sessions across different applications such as PPPoE, PPP, Layer 2 Tunneling Protocol (L2TP), and Subscriber Service Switch (SSS). The same session ID appears in the output for the **showpppoesession**, **showssssession**, and **showvpdnsession** commands.
PPPoE 3	PPPoE session ID.
Created	PPPoE session is created.
O PADS	Outgoing PPPoE Active Discovery Session-confirmation packet.
Connected PTA	PPPoE session is established.
Max session count(1) on mac(00b0.c2e9.c470) reached	PPPoE session is rejected because of per-MAC session limit.

Related Commands

Command	Description
encapsulation aal5autoppp virtual-template	Enables PPPoA/PPPoE autosense.
pppoe enable	Enables PPPoE sessions on an Ethernet interface or subinterface.
protocol pppoe (ATM VC)	Enables PPPoE sessions to be established on PVCs.
show pppoe session	Displays information about active PPPoE sessions.
show sss session	Displays Subscriber Service Switch session status.
show vpdn session	Displays session information about L2TP, L2F protocol, and PPPoE tunnels in a VPDN.

debug pppoe redundancy

To debug PPP over Ethernet (PPPoE) redundancy events on a dual Route Processor High Availability (HA) system and display cluster control manager (CCM) events and messages, use the **debug pppoe redundancy**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug pppoe redundancy
no debug pppoe redundancy

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(31)SB2	This command was introduced.
12.2(33)SRC	This command was integrated into Cisco IOS Release 12.2(33)SRC.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

The CCM provides the capability to facilitate and synchronize session initiation on the standby processor of a dual Route Processor HA system. Use the **debug pppoe redundancy** command to display CCM events and messages for PPPoE sessions.

Note This command is used only by Cisco engineers for internal debugging of CCM processes.

Examples

The following is sample output from the **debug pppoe redundancy** command from a Cisco 10000 series router active processor. No field descriptions are provided because command output is used for Cisco internal debugging purposes only.

```
Router# debug pppoe redundancy
Nov 22 17:21:11.327: PPPoE HA[0xBE000008] 9: Session ready to sync data
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = PADR, length = 58
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = SESSION ID, length = 2
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = SWITCH HDL, length = 4
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = SEGMENT HDL, length = 4
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = PHY SWIDB DESC, length = 20
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = VACCESS DESC, length = 28
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: Sync collection for ready events
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = PADR, length = 58
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = SESSION ID, length = 2
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = SWITCH HDL, length = 4
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = SEGMENT HDL, length = 4
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = PHY SWIDB DESC, length = 20
Nov 22 17:21:11.351: PPPoE HA[0xBE000008] 9: code = VACCESS DESC, length = 28
```

The following is sample output from the **debug pppoe redundancy** command from a Cisco 10000 series router standby processor:

```
Router# debug pppoe redundancy
Nov 22 17:21:11.448: PPPoE HA[0x82000008]: Recreating session: retrieving data
Nov 22 17:21:11.464: PPPoE HA[0x82000008] 9: Session ready to sync data
```

The following is sample output from the **debug pppoe redundancy** command from a Cisco 7600 series router active processor.

```
Router# debug pppoe redundancy
Dec 17 15:14:37.060: PPPoE HA[0x131B01B1] 28039: Session ready to sync data
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = PADR, length = 48
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = SESSION ID, length = 2
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = SWITCH HDL, length = 4
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = SEGMENT HDL, length = 4
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = PHY SWIDB DESC, length = 20
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = VACCESS DESC, length = 28
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: Sync collection for ready events
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = PADR, length = 48
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = SESSION ID, length = 2
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = SWITCH HDL, length = 4
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = SEGMENT HDL, length = 4
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = PHY SWIDB DESC, length = 20
Dec 17 15:14:37.076: PPPoE HA[0x131B01B1] 28039: code = VACCESS DESC, length = 28
```

The following is sample output from the **debug pppoe redundancy** command from a Cisco 7600 series router standby processor:

```
Router-stby# debug pppoe redundancy
Dec 17 15:14:37.180: STDBY: PPPoE HA[0xE41B019B]: Recreating session: retrieving data
Dec 17 15:14:37.204: STDBY: PPPoE HA[0xE41B019B] 28039: Session ready to sync data
```

debug presence

To display debugging information about the presence service, use the **debug presence**command in privileged EXEC mode. To disable debugging messages, use the **no** form of this command.

debug presence {**all** | **asnl** | **errors** | **event** | **info** | **timer** | **trace** | **xml**}
no debug presence {**all** | **asnl** | **errors** | **event** | **info** | **timer** | **trace** | **xml**}

Syntax Description

all	Displays all presence debugging messages.
asnl	Displays trace event logs in the Application Subscribe Notify Layer (ASNL).
errors	Displays presence error messages.
event	Displays presence event messages.
info	Displays general information about presence service.
timer	Displays presence timer information.
trace	Displays a trace of all presence activities.
xml	Displays messages related to the eXtensible Markup Language (XML) parser for presence service.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(11)XJ	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.

Examples

The following example shows output from the **debug presence asnl** command:

```
Router# debug presence asnl
*Sep  4 07:15:24.295: //PRESENCE:[17]:/presence_get_sccp_status: line is closed
*Sep  4 07:15:24.295: //PRESENCE:[17]:/presence_handle_line_update: line status changes,
send NOTIFY
*Sep  4 07:15:24.295: //PRESENCE:[17]:/presence_set_line_status: new line status [busy ]
*Sep  4 07:15:24.299: //PRESENCE:[17]:/presence_asnl_callback: type [5]
*Sep  4 07:15:24.299: //PRESENCE:[17]:/presence_asnl_callback: ASNL_RESP_NOTIFY_DONE
*Sep  4 07:15:24.299: //PRESENCE:[24]:/presence_get_sccp_status: line is closed
*Sep  4 07:15:24.299: //PRESENCE:[24]:/presence_handle_line_update: line status changes,
send NOTIFY
*Sep  4 07:15:24.299: //PRESENCE:[24]:/presence_set_line_status: new line status [busy ]
*Sep  4 07:15:24.299: //PRESENCE:[24]:/presence_asnl_callback: type [5]
*Sep  4 07:15:24.299: //PRESENCE:[24]:/presence_asnl_callback: ASNL_RESP_NOTIFY_DONE
*Sep  4 07:15:24.299: //PRESENCE:[240]:/presence_get_sccp_status: line is closed
*Sep  4 07:15:24.299: //PRESENCE:[240]:/presence_handle_line_update: line status changes,
```

```
send NOTIFY
*Sep  4 07:15:24.299: //PRESENCE:[240]:/presence_set_line_status: new line status [busy ]
*Sep  4 07:15:24.299: //PRESENCE:[766]:/presence_get_sccp_status: line is closed
*Sep  4 07:15:24.299: //PRESENCE:[766]:/presence_handle_line_update: line status changes,
send NOTIFY
*Sep  4 07:15:24.299: //PRESENCE:[766]:/presence_set_line_status: new line status [busy ]
*Sep  4 07:15:24.359: //PRESENCE:[766]:/presence_asnl_callback: type [5]
*Sep  4 07:15:24.359: //PRESENCE:[766]:/presence_asnl_callback: ASNL_RESP_NOTIFY_DONE
*Sep  4 07:15:24.811: //PRESENCE:[240]:/presence_asnl_callback: type [5]
*Sep  4 07:15:24.811: //PRESENCE:[240]:/presence_asnl_callback: ASNL_RESP_NOTIFY_DONE
*Sep  4 07:15:26.719: //PRESENCE:[17]:/presence_get_sccp_status: line is open
*Sep  4 07:15:26.719: //PRESENCE:[17]:/presence_handle_line_update: line status changes,
send NOTIFY
*Sep  4 07:15:26.719: //PRESENCE:[17]:/presence_set_line_status: new line status [idle ]
*Sep  4 07:15:26.719: //PRESENCE:[17]:/presence_asnl_callback: type [5]
*Sep  4 07:15:26.719: //PRESENCE:[17]:/presence_asnl_callback: ASNL_RESP_NOTIFY_DONE
*Sep  4 07:15:26.719: //PRESENCE:[24]:/presence_get_sccp_status: line is open
*Sep  4 07:15:26.719: //PRESENCE:[24]:/presence_handle_line_update: line status changes,
send NOTIFY
*Sep  4 07:15:26.719: //PRESENCE:[24]:/presence_set_line_status: new line status [idle ]
*Sep  4 07:15:26.723: //PRESENCE:[24]:/presence_asnl_callback: type [5]
*Sep  4 07:15:26.723: //PRESENCE:[24]:/presence_asnl_callback: ASNL_RESP_NOTIFY_DONE
```

The following example shows output from the **debug presence event**command:

```
Router# debug presence event
*Sep  4 07:16:02.715: //PRESENCE:[0]:/presence_sip_line_update: SIP nothing to update
*Sep  4 07:16:02.723: //PRESENCE:[17]:/presence_handle_notify_done: sip stack response code
 [29]
*Sep  4 07:16:02.723: //PRESENCE:[24]:/presence_handle_notify_done: sip stack response code
 [29]
*Sep  4 07:16:02.791: //PRESENCE:[240]:/presence_handle_notify_done: sip stack response
code [17]
*Sep  4 07:16:02.791: //PRESENCE:[766]:/presence_handle_notify_done: sip stack response
code [17]
*Sep  4 07:16:04.935: //PRESENCE:[0]:/presence_sip_line_update: SIP nothing to update
*Sep  4 07:16:04.943: //PRESENCE:[17]:/presence_handle_notify_done: sip stack response code
 [29]
*Sep  4 07:16:04.943: //PRESENCE:[24]:/presence_handle_notify_done: sip stack response code
 [29]
*Sep  4 07:16:04.995: //PRESENCE:[240]:/presence_handle_notify_done: sip stack response
code [17]
*Sep  4 07:16:04.999: //PRESENCE:[766]:/presence_handle_notify_done: sip stack response
code [17]
```

The following example shows output from the **debug presence info**command:

```
Router# debug presence info
*Sep  4 07:16:20.887: //PRESENCE:[17]:/presence_handle_line_update: get line status from
ccvdbPtr
*Sep  4 07:16:20.887: //PRESENCE:[17]:/presence_get_sccp_status: dn_tag 2
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <presence>
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <dm:person>
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <status>
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <e:activities>
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <tuple>
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <status>
```

```
*Sep  4 07:16:20.887: //PRESENCE:[16]:/presence_start_element_handler: line 1: unknown
element <e:activities>
*Sep  4 07:16:20.887: //PRESENCE:[0]:/presence_asnl_free_resp:
*Sep  4 07:16:20.887: //PRESENCE:[24]:/presence_handle_line_update: get line status from
ccvdbPtr
*Sep  4 07:16:20.887: //PRESENCE:[24]:/presence_get_sccp_status: dn_tag 2
*Sep  4 07:16:20.891: //PRESENCE:[23]:/presence_start_element_handler: line 1: unknown
element <presence>
```

The following example shows output from the **debug presence timer**command:

```
Router# debug presence timer
*Sep  4 07:16:41.271: //PRESENCE:[17]:/presence_asnl_notify_body_handler: expires time 3600
*Sep  4 07:16:41.271: //PRESENCE:[24]:/presence_asnl_notify_body_handler: expires time 3600
*Sep  4 07:16:41.271: //PRESENCE:[240]:/presence_asnl_notify_body_handler: expires time 607
*Sep  4 07:16:41.275: //PRESENCE:[766]:/presence_asnl_notify_body_handler: expires time 602
*Sep  4 07:16:43.331: //PRESENCE:[17]:/presence_asnl_notify_body_handler: expires time 3600
*Sep  4 07:16:43.331: //PRESENCE:[24]:/presence_asnl_notify_body_handler: expires time 3600
*Sep  4 07:16:43.331: //PRESENCE:[240]:/presence_asnl_notify_body_handler: expires time 605
*Sep  4 07:16:43.331: //PRESENCE:[766]:/presence_asnl_notify_body_handler: expires time 600
```

The following example shows output from the **debug presence trace**command:

```
Router# debug presence trace
*Sep  4 07:16:56.191: //PRESENCE:[17]:/presence_line_update:
*Sep  4 07:16:56.191: //PRESENCE:[24]:/presence_line_update:
*Sep  4 07:16:56.191: //PRESENCE:[240]:/presence_line_update:
*Sep  4 07:16:56.191: //PRESENCE:[766]:/presence_line_update:
*Sep  4 07:16:56.199: //PRESENCE:[17]:/presence_get_node_by_subid:
*Sep  4 07:16:56.199: //PRESENCE:[17]:/presence_handle_line_update:
*Sep  4 07:16:56.199: //PRESENCE:[17]:/presence_get_sccp_status:
*Sep  4 07:16:56.199: //PRESENCE:[17]:/presence_asnl_notify_body_handler:
*Sep  4 07:16:56.199: //PRESENCE:[24]:/presence_get_node_by_subid:
*Sep  4 07:16:56.199: //PRESENCE:[24]:/presence_handle_line_update:
*Sep  4 07:16:56.199: //PRESENCE:[24]:/presence_get_sccp_status:
*Sep  4 07:16:56.199: //PRESENCE:[24]:/presence_asnl_notify_body_handler:
*Sep  4 07:16:56.199: //PRESENCE:[240]:/presence_get_node_by_subid:
*Sep  4 07:16:56.199: //PRESENCE:[240]:/presence_handle_line_update:
*Sep  4 07:16:56.199: //PRESENCE:[240]:/presence_get_sccp_status:
*Sep  4 07:16:56.199: //PRESENCE:[240]:/presence_asnl_notify_body_handler:
*Sep  4 07:16:56.199: //PRESENCE:[766]:/presence_get_node_by_subid:
*Sep  4 07:16:56.203: //PRESENCE:[766]:/presence_handle_line_update:
*Sep  4 07:16:56.203: //PRESENCE:[766]:/presence_get_sccp_status:
*Sep  4 07:16:56.203: //PRESENCE:[766]:/presence_asnl_notify_body_handler:
*Sep  4 07:16:59.743: //PRESENCE:[17]:/presence_line_update:
*Sep  4 07:16:59.743: //PRESENCE:[24]:/presence_line_update:
*Sep  4 07:16:59.743: //PRESENCE:[240]:/presence_line_update:
*Sep  4 07:16:59.743: //PRESENCE:[766]:/presence_line_update:
```

The following example shows output from the **debug presence trace**command:

```
Router# debug presence trace
*Sep  4 07:17:17.351: //PRESENCE:[17]:/presence_xml_encode:
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_presence: keyword = presence
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_person: keyword = person
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_generic: keyword = Closed
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_activities: keyword = activities
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_otp: keyword = On-the-phone
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_tuple: keyword = tuple
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_status: keyword = status
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_generic: keyword = Closed
*Sep  4 07:17:17.355: //PRESENCE:[17]:/xml_encode_otp: keyword = On-the-phone
```

```
*Sep  4 07:17:17.355: <?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf" entity="sip:6003@1.4.171.34"
xmlns:e="urn:ietf:params:xml:ns:pidf:status:rpid"
xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model">
   <dm:person>
      <status>
         <basic>Closed</basic>
      </status>
      <e:activities>
         <e:on-the-phone/>
      </e:activities>
   </dm:person>
   <tuple id="cisco-cme">
      <status>
         <basic>Closed</basic>
         <e:activities>
            <e:on-the-phone/>
         </e:activities>
      </status>
   </tuple>
</presence>
```

Related Commands

Command	Description
presence	Enables presence service on the router and enters presence configuration mode.
presence enable	Allows the router to accept incoming presence requests.
show presence global	Displays configuration information about the presence service.
show presence subscription	Displays information about active presence subscriptions.

debug priority

To display priority queueing output, use the **debug priority**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug priority
no debug priority

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Examples

The following example shows how to enable priority queueing output:

```
Router# debug priority
Priority output queueing debugging is on
```

The following is sample output from the **debug priority** command when the Frame Relay PVC Interface Priority Queueing (FR PIPQ) feature is configured on serial interface 0:

```
Router# debug priority
00:49:05:PQ:Serial0 dlci 100 -> high
00:49:05:PQ:Serial0 output (Pk size/Q 24/0)
00:49:05:PQ:Serial0 dlci 100 -> high
00:49:05:PQ:Serial0 output (Pk size/Q 24/0)
00:49:05:PQ:Serial0 dlci 100 -> high
00:49:05:PQ:Serial0 output (Pk size/Q 24/0)
00:49:05:PQ:Serial0 dlci 200 -> medium
00:49:05:PQ:Serial0 output (Pk size/Q 24/1)
00:49:05:PQ:Serial0 dlci 300 -> normal
00:49:05:PQ:Serial0 output (Pk size/Q 24/2)
00:49:05:PQ:Serial0 dlci 400 -> low
00:49:05:PQ:Serial0 output (Pk size/Q 24/3)
```

Related Commands

Command	Description
debug custom-queue	Displays custom queueing output .

debug private-hosts

To enable debug messages for the Private Hosts feature, use the **debug private-hosts** command in privileged EXEC mode.

debug private-hosts {**all** | **events** | **acl** | **api**}

Syntax Description

all	Enable debug messages for all Private Hosts errors and events.
events	Enable debug messages for issues related to Private Hosts events.
acl	Enable debug messages for issues and events related to ACLs.
api	Enable debug messages for issues related to the application programming interface.

Command Default

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# debug private-hosts all

private-hosts events debugging is on
private-hosts api debugging is on
private-hosts acl debugging is on
Router#
```

Related Commands

Command	Description
debug fm private-hosts	Enables debug messages for the Private Hosts feature manager.

debug proxy h323 statistics

To enable proxy RTP statistics, use the **debug proxy h323 statistics**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug proxy h323 statistics
no debug proxy h323 statistics

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(2)NA	This command was introduced.

Usage Guidelines

Enter the **show proxy h323 detail-call** EXEC command to see the statistics.

debug pvcd

To display the permanent virtual circuit (PVC) Discovery events and Interim Local Management Interface (ILMI) MIB traffic used when discovering PVCs, use the **debug pvcd** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug pvcd
no debug pvcd

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command is primarily used by Cisco technical support representatives.

Examples

The following is sample output from the **debug pvcd** command:

```
Router# debug pvcd
PVCD: PVCD enabled w/ Subif
PVCD(2/0): clearing event queue
PVCD: 2/0 Forgetting discovered PVCs...
PVCD: Removing all dynamic PVCs on 2/0
PVCD: Restoring MIXED PVCs w/ default parms on 2/0
PVCD: Marking static PVCs as UNKNWN on 2/0
PVCD: Marking static PVC 0/50 as UNKNWN on 2/0 ...
PVCD: Trying to discover PVCs on 2/0...
PVCD: pvcd_discoverPVCs
PVCD: pvcd_ping
PVCD: fPortEntry.5.0 = 2
PVCD: pvcd_getPeerVccTableSize
PVCD: fLayerEntry.5.0 = 13
PVCD:end allocating VccTable size 13
PVCD:  pvcd_getPeerVccTable
PVCD:******* 2/0: getNext on fVccEntry = NULL TYPE/VALUE numFileds = 19 numVccs = 13
PVCD: Creating Dynamic PVC 0/33 on 2/0
PVCD(2/0): Before _update_inheritance() and _create_pvc() VC 0/33: DYNAMIC
PVCD: After _create_pvc() VC 0/33: DYNAMIC0/33 on 2/0 :  UBR PCR = -1
PVCD: Creating Dynamic PVC 0/34 on 2/0
PVCD(2/0): Before _update_inheritance() and _create_pvc() VC 0/34: DYNAMIC
PVCD: After _create_pvc() VC 0/34: DYNAMIC0/34 on 2/0 :  UBR PCR -1
PVCD: Creating Dynamic PVC 0/44 on 2/0
PVCD(2/0): Before _update_inheritance() and _create_pvc() VC 0/44: DYNAMIC
PVCD: After _create_pvc() VC 0/44: DYNAMIC0/44 on 2/0 :  UBR PCR = -1
PVCD: PVC 0/50 with INHERITED_QOSTYPE
PVCD: _oi_state_change ( 0/50, 1 = ILMI_VC_UP )
PVCD: Creating Dynamic PVC 0/60 on 2/0
PVCD(2/0): Before _update_inheritance() and _create_pvc() VC 0/60: DYNAMIC
PVCD: After _create_pvc() VC 0/60: DYNAMIC0/60 on 2/0 :  UBR PCR = -1
PVCD: Creating Dynamic PVC 0/80 on 2/0
PVCD(2/0): Before _update_inheritance() and _create_pvc() VC 0/80: DYNAMIC
PVCD: After _create_pvc() VC 0/80: DYNAMIC0/80 on 2/0 :  UBR PCR = -1
PVCD: Creating Dynamic PVC 0/99 on 2/0
```

debug pvdm2dm

To view contents of packets flowing through PVDMII-xxDM digital modem devices, use the **debug pvdm2dm** command in privileged EXEC mode. To disable debug activity, use the **no** form of this command.

debug pvdm2dm {**packet modem** | **pvdm** *slot*/*port* | *pvdm slot*}
no debug pvdm2dm

Syntax Description

packet	Debugs packets
modem	Debugs modem packets
pvdm	Debugs PVDM packets
slot	Router slot for pvdms/modems
port	Modem number
pvdm slot	PVDM number

Command Default

Disabled

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

To debug the contents of modem packets for a specific modem, use the following command:

- **debug pvdm2dm packet modem** *<slot>/<port>*

By removing the specific modem number at the end, one can enable packet debugging for all the modems available on the router:

- **debug pvdm2dm packet modem**

The following command enables packet debugging for all packets flowing through a particular PVDMII-xxDM device:

- **debug pvdm2dm packet pvdm** *<slot>/<pvdm slot>*

The following command enables debugging of packets flowing through any PVDMII-xxDM device:

- **debug pvdm2dm packet pvdm**

The following command enables debugging of packets flowing through any PVDMII-xxDM device and any PVDMII-xxDM-based modem channel:

- **debug pvdm2dm packet**

To see what debug flags are set, and to view the contents of debugged packets, use the **show debugging** command.

Examples

The following example sets debugging for a specific modem. The following **show debugging** command displays the debug flag that is set, and gives a typical printout for one debugged packet:

```
Router# debug pvdm2dm packet modem 0/322
Router# show debugging
PVDM2 DM:
  Modem 0/322 packet debugging is on
Router#
May 24 17:35:16.318: pvdm2_dm_tx_dsp_pak_common: bay 0, dsp 0 May 24 17:35:16.318:
pvdm2_dm_dump_pak_hex: pak: 43E1F6FC size 8 May 24 17:35:16.318: 00 08 00 00 00 1C 00 00
May 24 17:35:16.322:
```

The following example sets debugging for all PVDMII-xxDM modems available on the router.

```
Router# debug pvdm2dm packet
Router# show debugging
PVDM2 DM:
  Modem 0/322 packet debugging is on
  Modem 0/323 packet debugging is on
  Modem 0/324 packet debugging is on
  .
  .
  .
  Modem 0/355 packet debugging is on
  Modem 0/356 packet debugging is on
  Modem 0/357 packet debugging is on
Router#
```

The following example sets debugging for a particular PVDMII-xxDM device.

```
Router# debug pvdm2dm packet pvdm 0/0
Router# show debugging
PVDM2 DM:
  PVDM2 0/0 packet debugging is on
Router#
```

The following example sets debugging for all PVDMII-xxDM devices in the router.

```
Router# debug pvdm2dm packet pvdm
Router# show debugging
PVDM2 DM:
  PVDM2 0/0 packet debugging is on
  PVDM2 0/1 packet debugging is on
  PVDM2 0/2 packet debugging is on
Router#
```

In all of these examples, the output describing the debugged packets is similar to that of the first example, except that the packet contents will vary.

Related Commands

Command	Description
show debugging	Displays information about the type of debugging enabled for your router.

debug pw-udp

To debug pseudowire User Datagram Protocol (UDP) virtual circuits (VCs), use the **debug pw-udp**command in privileged EXEC mode.

debug pw-udp {**errors** | **events** | **fsm**}

Syntax Description

errors	Specifies pseudowire UDP errors.
events	Specifies pseudowire UDP events.
fsm	Specifies pseudowire UDP finite state machine (FSM).

Command Default

Debugging for pseudowire UDP VCs is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(2)S	This command was introduced.

Usage Guidelines

To debug pseudowire UDP VCs, you must configure the **debug pw-udp**command in conjunction with the following set of **debug** commands before configuring Circuit Emulation Service over UDP (CESoUDP):

On both active and standby route processors (RPs):

- **debug xconnect event**
- **debug xconnect error**
- **debug acircuit event**
- **debug acircuit error**
- **debug acircuit checkpoint**
- **debug pw-udp checkpoint**
- **debug ssm cm events**
- **debug ssm cm errors**
- **debug ssm sm errors**
- **debug ssm sm events**
- **debug sss error**
- **debug sss event**
- **debug sss fsm**
- **debug cem ac event**
- **debug cem ac error**

- **debug cem ha event**
- **debug cem ha error**

On the Circuit Emulation over Packet (CeOP) line card:

- **debug ssm cm events**
- **debug ssm cm errors**
- **debug ssm sm errors**
- **debug ssm sm events**

For more information about each of these debug commands, see the Cisco IOS Debug Command Reference Guide.

Examples

The following example shows how to debug pseudowire UDP VCs on the active RP:

```
Router#
debug xconnect event
Xconnect author event debugging is on
Router#
debug xconnect error
Xconnect author errors debugging is on
Router#
debug acircuit event
Attachment Circuit events debugging is on
Router#
debug acircuit error
Attachment Circuit errors debugging is on
Router#
debug cem ac event
CEM AC Events debugging is on
Router#
debug cem ac error
CEM AC Error debugging is on
Router#
debug cem ha event
CEM redundancy events debugging is on
Router#
debug cem ha error
CEM redundancy error debugging is on
Router#
debug pw-udp event
PW UDP events debugging is on
Router#
debug pw-udp error
PW UDP errors debugging is on
Router#
debug pw-udp fsm
PW UDP fsm debugging is on
Router#
debug ssm cm events
SSM Connection Manager events debugging is on
Router#
debug ssm cm errors
SSM Connection Manager errors debugging is on
Router#
debug ssm sm errors
```

```
SSM Segment Handler Manager errors debugging is on
Router#
debug ssm sm events
SSM Segment Handler Manager events debugging is on
Router#
debug sss error
SSS Manager errors debugging is on
Router#
debug sss event
SSS Manager events debugging is on
Router#
debug sss fsm
SSS Manager fsm debugging is on
Router#
00:05:01: STDBY: CEMHA RF: CID 116, Seq 219, Event RF_EVENT_CLIENT_PROGRESSION, Op 7, State
 STANDBY COLD-BULK, Peer ACTIVE
00:05:01: STDBY: CEMHA CF: CF client 182, entity 0 received msg
00:05:01: STDBY: CEMHA CF: CF client 182, entity 0 received msg
00:05:01: STDBY: CEMHA CF: CF client 182, entity 0 received msg
00:05:01: STDBY: CEMHA CF: CF client 182, entity 0 received msg
00:05:01: STDBY: CEMHA CF: CF client 182, entity 0 received msg
00:05:01: STDBY: CEMHA CF: CF client 182, entity 0 recei
00:05:01: STDBY: CEMHA CF Received Interface Update event=0x10
00:05:01: STDBY: AC CESP[CE4/2/0]: Activated CEM group 0
00:05:01: STDBY: AC CESP[CE4/2/0]: Setup switching of ckt 0
00:05:01: STDBY: AC CESP ERROR[CE4/2/0]: (CEM4/2/0): Setup Switching 0 cannot proceed sw/seg:
 0/0, Flag 10, SSM 0
00:05:01: STDBY: AC CESP ERROR[CE4/2/0]: CEM 0 Data switching setup failed
00:05:01: STDBY: CEMHA CF Received T1/E1 Update event=0x20
00:05:01: STDBY: CEMHA CF Received Interface Update event=0x10
00:05:01: STDBY: AC CESP[CE4/2/1]: Activated CEM group 0
00:05:01: STDBY: AC CESP[CE4/2/1]: Setup switching of ckt 0
00:05:01: STDBY: AC CESP ERROR[CE4/2/1]: (CEM4/2/1): Setup Switching 0 cannot proceed sw/seg:
 0/0, Flag 10, SSM 0
00:05:01: STDBY: AC CESP ERROR[CE4/2/1]: CEM 0 Data switching setup failed
00:05:01: STDBY: CEMHA CF Received T1/E1 Update event=0x20
00:05:01: STDBY: CEMHA(CEM4/2/1):Decode received VC AC for evtype 8 cem_id = 0,
 pw_state = 1, seg 3007, switch 2002, ac_wait_flags = 10 ,is_standby = NO, red_seg 0,
red_switch 0
00:05:01: STDBY: CEMHA: cem_id0, before decode sw/segment: 0/0, seg_state = 2, red sw/segment:
 0/0
00:05:01: STDBY: SSM SM ID LOCK: [CEM HA:id_lock_util_init:0] locker <ALL>: instance created
 for <SSM SM ID LOCK>
00:05:01: STDBY: SSM CM[12295]: reserve ID: Locking SSM ID
00:05:01: STDBY: SSM SM ID LOCK: [CEM HA:id_lock:12295] locker <SIP>: count 0 --> 1
00:05:01: STDBY: CEMHA CF Received Interface Update event=0x10
00:05:01: STDBY: AC CESP[CE4/2/1]: Activated CEM group 0
00:05:01: STDBY: CEMHA CF Received T1/E1 Update event=0x20
00:05:01: STDBY: CEMHA CF Received Interface Update event=0x10
00:05:01: STDBY: AC CESP[CE4/2/1]: Activated CEM group 0
00:05:01: STDBY: CEMHA CF Received T1/E1 Update event=0x20
00:05:01: STDBY: CEMHA CF: Received bulk sync complete - sending ack
00:05:01: STDBY: CEMHA: Create CEM Circuit verification Background process...
00:05:01: STDBY: SSM CM: reserve seg(12295) sw(8194) IDs
00:05:01: STDBY: CEMHA : CEM HA Background Process
00:05:02: STDBY: CEMHA: CF sync successfully completed
00:05:03: STDBY: XCL2 CID 119 Seq 224 Event RF_EVENT_CLIENT_PROGRESSION Op 7 State STANDBY
 COLD-BULK Peer ACTIVE
00:05:03: STDBY: PW UDP HA: HA Coexistence. Skip ISSU Negotiation on standby RP
00:05:06: STDBY: CEM HA: (CEM4/2/0) CEM 0x0 Platform chkpt data has
                      arrived for cktid=0
00:05:06: STDBY: CEM PW: Remove from WaitQ, ckt_type 19
00:05:06: STDBY: CEM HA: (CEM4/2/1) CEM 0x0 Platform chkpt data has
                      arrived for cktid=0
```

```
00:05:06: STDBY: CEM PW: Remove from WaitQ, ckt_type 19
00:05:06: STDBY: AC CESP[CE4/2/1]: Setup switching of ckt 0
00:05:06: STDBY: AC: [CE4/2/1, 0]: Setup switching
00:05:06: STDBY: AC: [CE4/2/1, 0]: Our AIE EF000002 Peer's AIE 2B000004 Peer's peer 00000000
00:05:06: STDBY: AC: [CE4/2/1, 0]: Using switch hdl 8194
00:05:06: STDBY: SSM CM[12295]: provision segment: standby RP received existing id from
active RP
00:05:06: STDBY: AC: [CE4/2/1, 0]: Successfully setup switching API
00:05:06: STDBY: AC: [CE4/2/1, 0]: Allocated segment hdl 12295
00:05:06: STDBY: AC CESP[CE4/2/1]: CKT UP ID: 0
00:05:06: STDBY: AC CESP[CE4/2/1]: Send ACMGR NOTIF, ckt_type 19, ckt_id 0 UP
00:05:06: STDBY: AC: Update seg 12295 plane with circuit Up status
00:05:06: STDBY: SSM SH[12295]: X: alloc sbase 0x500386A0 hdl 3007
00:05:06: STDBY: SSM CM[12295]: [CESoPSN Basic] provision first allocated base now, reserved
 earlier
00:05:06: STDBY: SSM CM[12295]: CM FSM: st Idle, ev Prov seg->Down
00:05:06: STDBY: SSM SH[12295]: init segment base
00:05:06: STDBY: SSM SH[ADJ:CESoPSN Basic:12295]: init segment class
00:05:06: STDBY: SSM CM[ADJ:CESoPSN Basic:12295]: provision segment 1
00:05:06: STDBY: SSM SM[ADJ:CESoPSN Basic:12295]: Provision segment: Idle -> Prov
00:05:06: STDBY: SSM SM[ADJ:CESoPSN Basic:12295]: provision segment
00:05:06: STDBY: SSM CM[12295]: segment status update Up
00:05:06: STDBY: SSM CM[12295]: CM FSM: st Down, ev Upd seg->Down
00:05:06: STDBY: SSM CM[ADJ:CESoPSN Basic:12295]: update segment status
00:05:06: STDBY: SSM SM[ADJ:CESoPSN Basic:12295]: Update segment: no state change, Prov
00:05:06: STDBY: SSM ADJ[ADJ:CESoPSN Basic:CE4/2/1: Type L:12295]: update segment status:
Up
00:05:06: STDBY: SSM ADJ[ADJ:CESoPSN Basic:CE4/2/1: Type L:12295]: ATM Async is supported
00:05:06: STDBY: SSM ADJ[ADJ:CESoPSN Basic:CE4/2/1: Type L:12295]: Platform requesting not
 to send unready: 1
00:05:06: STDBY: SSM ADJ[ADJ:CESoPSN Basic:CE4/2/1: Type L:12295]: circuit Up event
00:05:06: STDBY: SSM ADJ[ADJ:CESoPSN Basic:CE4/2/1: Type L:12295]: send segment ready
00:05:06: STDBY: SSM CM[12295]: [ADJ] shQ request send ready event
00:05:06: STDBY: ACMGR [CE4/2/1]: Receive <Circuit Status> msg
00:05:06: STDBY: ACMGR [CE4/2/1]: circuit up event, FSP state chg sip up to both up, action
 is peer p2p up, circuit remote up
00:05:06: STDBY: SSS MGR [uid:4]: Handling peer-to-peer event
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: receive p2p msg type: circuit status
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: Success to obtain circuit type 19 from AC
Access IE
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: event local ac up, state changed from
provisioned to activating, action local_ac_up
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: Waiting for vc checkpoint data
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: Success to obtain circuit type 19 from AC
Access IE
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: event need checkpoint, state changed from
activating to checkpoint wait, action clean_up_checkpoint_resource
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: Cleanup Checkpoint Resources
00:05:06: STDBY: PW UDP MGR [10.1.1.153, 200]: local ac status is changed from none to UP
00:05:06: STDBY: SSM CM[12295]: SM msg event send ready event
00:05:06: STDBY: SSM SM[ADJ:CESoPSN Basic:12295]: segment ready
00:05:06: STDBY: SSM SM[ADJ:CESoPSN Basic:12295]: Found segment data: Prov -> Ready
00:05:07: STDBY: CEMHA RF: CID 116, Seq 219, Event RF_EVENT_CLIENT_PROGRESSION, Op 8, State
 STANDBY HOT, Peer ACTIVE
00:05:07: STDBY: XCL2 CID 119 Seq 224 Event RF_EVENT_CLIENT_PROGRESSION Op 8 State STANDBY
 HOT Peer ACTIVE
00:05:07: STDBY: PW UDP HA: HA Coexistence. Skip ISSU Negotiation on standby RP
```

Related Commands

Command	Description
encapsulation (pseudowire)	Specifies an encapsulation type for tunneling Layer 2 traffic over a pseudowire.

Command	Description
udp port	Configures the UDP port information on the xconnect class.
show pw-udp vc	Displays information about pseudowire UDP VCs.

debug pxf atom

To display debug messages relating to Parallel eXpress Forwarding (PXF) Any Transport over MPLS (AToM), use the debug pxf atom command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf atom [{**ac** | **mpls**}]
no debug pxf atom [{**ac** | **mpls**}]

Syntax Description

ac	(Optional) Displays AToM information related to attachment circuit (AC) events.
mpls	(Optional) Displays AToM information related to MPLS Forwarding Infrastructure (MFI) events.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF AToM AC events debug messages:

```
Router# debug pxf atom ac
PXF ATOM AC debugging is on
```

Related Commands

Command	Description
show mpls l2transport	Displays information about AToM virtual circuits (VCs) that have been enabled to route Layer 2 packets on a router, including platform-independent AToM status and capabilities of a particular interface.
show mpls l2transport vc	Displays information about AToM VCs that are enabled to route Layer 2 packets on a router.
show pxf cpu atom	Displays PXF AToM information for an interface or VCCI.
show pxf cpu mpls label	Displays PXF forwarding information for a label.
show pxf cpu statistics atom	Displays PXF CPU AToM statistics.

debug pxf backwalks

To display debug messages relating to Parallel eXpress Forwarding (PXF) backwalk requests, use the debug pxf backwalks command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf backwalks
no debug pxf backwalks

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF backwalk requests debug messages:

```
Router# debug pxf backwalks
PXF BACKWALK debugging is on
```

Related Commands

Command	Description
show pxf cpu statistics backwalk	Displays PXF CPU backwalk requests statistics.

debug pxf bba

To display debug messages relating to Parallel eXpress Forwarding (PXF) Broadband Access Aggregation (BBA) features, use the debug pxf bba command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf bba [{**ac sh_counter** | **ac sh_error** | **ac sh event** | **elog** | **l2f startstop debug** | **l2x fh error** | **l2x fh event** | **l2x sh counter** | **l2x sh error** | **l2x sh event** | **lt sh error** | **lt sh event**}]
no debug pxf bba [{**ac sh_counter** | **ac sh_error** | **ac sh event** | **elog** | **l2f startstop debug** | **l2x fh error** | **l2x fh event** | **l2x sh counter** | **l2x sh error** | **l2x sh event** | **lt sh error** | **lt sh event**}]

Syntax Description

ac_sh_counter	(Optional) Displays attachment circuit (AC) segment counters.
ac_sh_error	(Optional) Displays AC segment errors.
ac_sh_event	(Optional) Displays AC segment events.
elog	(Optional) Displays event logging messages.
l2f_startstop_debug	(Optional) Displays Layer 2 Forwarding (L2F) tunneling events.
l2x_fh_error	(Optional) Displays L2F/L2TP (L2x) feature errors.
l2x_fh_event	(Optional) Displays L2x feature events.
l2x_sh_counter	(Optional) Displays L2x segment counters.
l2x_sh_error	(Optional) Displays L2x segment errors.
l2x_sh_event	(Optional) Displays L2x segment events.
lt_sh_error	(Optional) Displays LT segment errors.
lt_sh_event	(Optional) Displays LT segment events.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display AC segment counters debug messages:

```
Router# debug pxf bba ac_sh_counter
AC segment counters debugging is on
*Jan 19 13:18:26.698: c10k_get_ac_segment_counters: pppox vcci 2709 rx pkts = 0
rx byte = 0 tx pkts = 0 tx bytes = 0
*Jan 19 13:18:26.698: c10k_get_ac_segment_counters: pppox vcci 2709 tx drop pkts
 = 0 tx drop bytes = 0
```

```
*Jan 19 13:18:26.698: c10k_get_ac_segment_counters: pppox vcci 2710 rx pkts = 0
rx byte = 0 tx pkts = 0 tx bytes = 0
*Jan 19 13:18:26.698: c10k_get_ac_segment_counters: pppox vcci 2710 tx drop pkts
 = 0 tx drop bytes = 0
*Jan 19 13:18:36.698: c10k_get_ac_segment_counters: pppox vcci 2709 rx pkts = 0
rx byte = 0 tx pkts = 0 tx bytes = 0
*Jan 19 13:18:36.698: c10k_get_ac_segment_counters: pppox vcci 2709 tx drop pkts
 = 0 tx drop bytes = 0
*Jan 19 13:18:36.698: c10k_get_ac_segment_counters: pppox vcci 2710 rx pkts = 0
rx byte = 0 tx pkts = 0 tx bytes = 0
.
.
.
```

The following example shows how to display L2F tunneling debug messages:

```
Router# debug pxf bba l2f_startstop_debug
L2F feature debugging is on
*Jan 20 12:04:18.976: hwcnts.rx_pkts :0 hwcnts.rx_bytes :0
hwcnts.tx_pkts :0 hwcnts.tx_bytes: 0
hwcnts.tx_drop_pkts :0 hwcnts.tx_drop_bytes: 0
pcntrs->rx_pkts: 0 pcntrs->rx_bytes: 0
pcntrs->tx_pkts: 0 pcntrs->tx_bytes: 0
pcntrs->tx_drop_pkts: 0 pcntrs->tx_drop_bytes: 0
*Jan 20 12:04:18.976: hwcnts.rx_pkts :0 hwcnts.rx_bytes :0
hwcnts.tx_pkts :0 hwcnts.tx_bytes: 0
hwcnts.tx_drop_pkts :0 hwcnts.tx_drop_bytes: 0
pcntrs->rx_pkts: 0 pcntrs->rx_bytes: 0
pcntrs->tx_pkts: 0 pcntrs->tx_bytes: 0
pcntrs->tx_drop_pkts: 0 pcntrs->tx_drop_bytes: 0
.
.
.
```

Related Commands

Command	Description
show pxf cpu bba	Displays PXF CPU (RP) BBA information.

debug pxf cef

To display debug messages relating to Parallel eXpress Forwarding (PXF) Cisco Express Forwarding (CEF), use the debug pxf cef command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf cef [{**fibroot** | **rpf**}]
no debug pxf cef [{**fibroot** | **rpf**}]

Syntax Description

fibroot	Displays PXF CEF Forwarding Information Base (FIB) root information.
rpf	Displays PXF CEF Reverse Path Forwarding (RPF) information.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF CEF debug messages:

```
Router# debug pxf cef
PXF CEF debugging is on
```

Related Commands

Command	Description
show ip cef	Displays summary information about the FIB entries.
show pxf cpu cef	Displays PXF CPU memory usage, CEF, and External Column Memory (XCM) information.

debug pxf dma

To display debug messages relating to Parallel eXpress Forwarding (PXF) direct memory access (DMA) operations, use the debug pxf dma command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf dma
no debug pxf dma

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)XI	This command was introduced.
12.2(31)SB	This command was integrated into Cisco IOS Release 12.2(31)SB.

Examples

The following example shows how to display PXF DMA ASIC debug messages:

```
Router# debug pxf dma
PXF DMA ASIC debugging is on
*Jan  4 08:05:06.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:06.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:07.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:07.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:08.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:08.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:09.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:09.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:10.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:10.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:11.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:11.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:12.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:12.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:13.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:13.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:14.314: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:14.814: get ftbb reg: slot 3, subslot 1
*Jan  4 08:05:14.982: Entering c10k_cobalt_send.
*Jan  4 08:05:14.982:   Packet decode: datagramstart 0x0A0301BE length 76
*Jan  4 08:05:14.982:   Header decode: Chan 0, VCCI 2515
*Jan  4 08:05:14.982:   Header decode: flags 0x0001
*Jan  4 08:05:14.982:   c10k_cobalt_send: Checked the idb state.
*Jan  4 08:05:14.982:   c10k_cobalt_send: Checked the FromRP Q count.
.
.
.
```

Related Commands

Command	Description
show pxf dma	Displays the current state of the DMA buffers, error counters, and registers on the PXF.

debug pxf iedge

To display debug messages relating to Parallel eXpress Forwarding (PXF) Intelligent Edge (iEdge) operations, use the debug pxf iedge command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf iedge [**stats**]
no debug pxf iedge [**stats**]

Syntax Description

stats	(Optional) Includes PXF iEdge statistics in the output.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF iEdge debug messages:

```
Router# debug pxf iedge
iEdge Feature Debug debugging is on
```

Related Commands

Command	Description
show pxf cpu iedge	Displays PXF iEdge information for an interface or policy.

debug pxf ipv6

To display debug messages relating to Parallel eXpress Forwarding (PXF) IPv6 provisioning, use the debug pxf ipv6 command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf ipv6 [{**acl** | **fib** | **hash**}]
no debug pxf ipv6 [{**acl** | **fib** | **hash**}]

Syntax Description

acl	(Optional) Displays PXF IPv6 access control list (ACL) information.
fib	(Optional) Displays PXF Forwarding Information Base (FIB) information.
hash	(Optional) Displays PXF IPv6 hash information.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF IPv6 ACL debug messages:

```
Router# debug pxf ipv6 acl
PXF IPV6 ACL debugging is on
```

Related Commands

Command	Description
show ipv6 interface	Displays IPv6 interface settings.
show ipv6 route	Displays IPv6 routing table contents.
show pxf cpu ipv6	Displays PXF CPU IPv6 statistics.

debug pxf l2less-error

To display debug messages relating to Parallel eXpress Forwarding (PXF) Layer 2 Less (L2less) drop packet errors, use the debug pxf l2less-error command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf l2less-error
no debug pxf l2less-error

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)XI	This command was introduced.

Usage Guidelines

The Route Processor (RP) uses the L2less packet handler to handle tunneling encapsulated packets that do not have the original IP and Layer 2 information associated with them. The L2less handler takes the packet with a specific header, updates the statistics (interface packet and byte counts), and enqueues the packet to the IP input queue.

Examples

The following example shows how to display PXF L2less drop packet errors debug messages:

```
Router# debug pxf l2less-error
PXF l2less-error debugging is on
```

Related Commands

Command	Description
show pxf statistics	Displays chassis-wide, summary PXF statistics.

debug pxf microcode

To display debug message relating to Parallel eXpress Forwarding (PXF) microcode operations, use the debug pxf microcode command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf microcode
no debug pxf microcode

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)XI	This command was introduced.

Examples

The following example shows how to display PXF microcode debug messages:

```
Router# debug pxf microcode
PXF microcode debugging is on
```

Related Commands

Command	Description
microcode reload	Reloads the Cisco IOS image from a line card on a Cisco router.
show pxf microcode	Displays identifying information for the microcode currently loaded on the PXF.

debug pxf mnode

To display debug messages relating to Parallel eXpress Forwarding (PXF) multiway node (mnode) operations, use the debug pxf mnode command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf mnode
no debug pxf mnode

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Usage Guidelines

The mnodes are used in the multiway tree (Mtrie) library. Each mnode has a number of buckets that point to lower level mnodes or to multiway leaves (mleaves). The mleaves can be null leaves which indicate empty buckets.

Examples

The following example shows how to display PXF mnode debug messages:

```
Router# debug pxf mnode
PXF MNODE debugging is on
```

Related Commands

Command	Description
show pxf cpu cef	Displays PXF CPU memory usage, Cisco Express Forwarding, and XCM information.

debug pxf mpls

To display debug messages relating to Parallel eXpress Forwarding (PXF) Multiprotocol Label Switching (MPLS) operations, use the debug pxf mpls command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf mpls [{**csc** {**event** | **stats**} | **lspv**}]
no debug pxf mpls [{**csc** {**event** | **stats**} | **lspv**}]

Syntax Description

csc {**event** | **stats**}	(Optional) Displays PXF Cisco Signaling Controller (CSC) events and statistics.
lspv	Displays Link State Path Vector (LSPV) debug messages from the PXF MPLS Label Switched Path (LSP) Ping/Traceroute feature.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF MPLS CSC statistics debug messages:

```
Router# debug pxf mpls csc stats
PXF MPLS CSC STATS debugging is on
```

Related Commands

Command	Description
ping mpls	Checks MPLS LSP connectivity.
show mpls interfaces	Displays information about the interfaces that have been configured for label switching.
show pxf cpu mpls	Displays PXF MPLS (FIB) entry information.
trace mpls	Discovers MPLS LSP routes that packets will take when traveling to their destinations.

debug pxf mroute

To display debug messages relating to Parallel eXpress Forwarding (PXF) multicast route (mroute) operations, use the debug pxf mroute command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf mroute [{**mdb** | **mdt** | **midb** | **punt**}]
no debug pxf mroute [{**mdb** | **mdt** | **midb** | **punt**}]

Syntax Description

mdb	(Optional) Displays PXF multicast descriptor block (MDB) event messages.
mdt	(Optional) Displays PXF multicast distribution tree (MDT) messages.
midb	(Optional) Displays PXF multicast interface descriptor block (MIDB) messages.
punt	(Optional) Displays PXF multicast punted packets information.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF multicast distribution tree (MDT) debug messages:

```
Router# debug pxf mroute mdt
PXF mroute mdt creation debugging is on
```

Related Commands

Command	Description
clear ip mroute	Deletes entries from the IP multicast routing table.
show ip mroute	Displays the contents of the IP multicast routing table.
show pxf cpu mroute	Displays PXF multicast routing information for a particular group or range of groups.

debug pxf multilink

To display debug messages relating to Parallel eXpress Forwarding (PXF) multilink operations, use the debug pxf multilink command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf multilink [{**all** | **atm** | **frame-relay** | **frfl2** | **lfi** | **ppp** | **queue** | **rates**}]
no debug pxf multilink [{**all** | **atm** | **frame-relay** | **frfl2** | **lfi** | **ppp** | **queue** | **rates**}]

Syntax Description

all	(Optional) Displays all PXF multilink messages.
atm	(Optional) Displays PXF multilink ATM messages.
frame-relay	(Optional) Displays PXF multilink Frame Relay messages.
frf12	(Optional) Displays PXF Frame Relay Forum FRF.12-based fragmentation information on Frame Relay permanent virtual circuits (PVCs).
lfi	(Optional) Displays PXF Link Fragmentation and Interleaving (LFI) messages.
ppp	(Optional) Displays PXF multilink PPP messages.
queue	(Optional) Displays PXF multilink queue messages.
rates	(Optional) Displays PXF multilink queue rate messages.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF multilink ATM debug messages:

```
Router# debug pxf multilink atm
Router#
```

Related Commands

Command	Description
frame-relay fragment	Enables fragmentation of Frame Relay frames on a Frame Relay map class.
show ppp multilink	Displays bundle information for the MLP bundles.
show pxf statistics	Displays chassis-wide, summary PXF statistics.

debug pxf netflow

To enable debugging of NetFlow Parallel eXpress Forwarding (PXF)operations, use the debug pxf netflow command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf netflow {**records** | **time**}
no debug pxf netflow {**records** | **time**}

Syntax Description

records	Displays NetFlow PXF records information.
time	Displays NetFlow PXF time synchronization information.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)XI	This command was introduced.

Examples

The following example enables NetFlow PXF records debugging:

```
Router# debug pxf netflow records
PXF netflow records debugging is on
Router#
```

Related Commands

Command	Description
show pxf netflow	Displays NetFlow PXF counters information.

debug pxf pbr

To display debug messages relating to Parallel eXpress Forwarding (PXF) policy-based routing (PBR), use the debug pxf pbr command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf pbr [{**sacl** | **trace**}]
no debug pxf pbr [{**sacl** | **trace**}]

Syntax Description

sacl	(Optional) Displays PXF PBR super access control list (ACL) messages.
trace	(Optional) Displays PXF PBR trace information.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF PBR trace debug messages:

```
Router# debug pxf pbr trace
PXF PBR Trace debugging is on
```

Related Commands

Command	Description
show pxf cpu pbr action	Displays the PBR actions configured on the PXF for all PBR route maps.

debug pxf qos

To display debug messages relating to Parallel eXpress Forwarding (PXF) quality of service (QoS) operations, use the debug pxf qos command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf qos [{**ipc** | **trace**}]
no debug pxf qos [{**ipc** | **trace**}]

Syntax Description

ipc	(Optional) Displays PXF QoS interprocess communication (IPC) information.
trace	(Optional) Displays PXF QoS trace information

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2S	This command was introduced.

Examples

The following example shows how to display PXF QoS IPC debug messages:

```
Router# debug pxf qos trace
PXF QoS IPC Events debugging is on
Router#
*Apr 30 23:23:44:  c10k_bandwidth_notification_handler: cmdtype=4 event=0x30 acA
*Apr 30 23:23:44:  c10k_priority_notification_handler: cmdtype=4 event=0x30 actA
*Apr 30 23:23:44:  c10k_bandwidth_notification_handler: cmdtype=4 event=0x30 acA
*Apr 30 23:23:44:  c10k_bandwidth_notification_handler: cmdtype=4 event=0x30 acA
*Apr 30 23:23:44:  c10k_priority_notification_handler: cmdtype=4 event=0x30 actA
*Apr 30 23:23:44:  c10k_bandwidth_notification_handler: cmdtype=4 event=0x30 acA
.
.
.
```

Related Commands

Command	Description
show pxf cpu qos	Displays External Column Memory (XCM) contents related to a particular policy.
show pxf statistics	Displays chassis-wide, summary PXF statistics.

debug pxf stats

To display debug messages relating to Parallel eXpress Forwarding (PXF) statistics collector events, use the debug pxf stats command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf stats
no debug pxf stats

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)XI	This command was introduced.

Examples

The following example shows how to display PXF statistics debug messages:

```
Router# debug pxf stats
PXF hardware statistics debugging is on
```

Related Commands

Command	Description
clear pxf	Clears PXF counters and statistics.
show pxf cpu statistics	Displays PXF CPU statistics.
show pxf statistics	Displays chassis-wide, summary PXF statistics.

debug pxf subblocks

To display debug messages relating to Parallel eXpress Forwarding (PXF) bridged subinterfaces (encapsulation types), use the debug pxf subblocks command in privileged EXEC mode. To disable the debugging, use the no form of this command.

debug pxf subblocks
no debug pxf subblocks

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled (debugging is not enabled).

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)XI	This command was introduced.

Examples

The following example shows how to display PXF bridged subinterfaces (encapsulation type) debug messages:

```
Router# debug pxf subblocks
PXF hardware subblock debugging is on
```

Related Commands

Command	Description
show pxf cpu statistics	Displays PXF CPU statistics.
show pxf cpu subblocks	Displays PXF CPU statistics for bridged subinterfaces (encapsulation types).

debug pxf tbridge

To enable debugging of Parallel eXpress Forwarding (PXF) transparent bridging, use the **debug pxf tbridge**command in privileged EXEC mode. To disable debugging for the PXF transparent bridge, use the **no** form of this command.

debug pxf tbridge
no debug pxf tbridge

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(31)SB	This command was integrated into Cisco IOS Release 12.2(31)SB and implemented on the Cisco 10000 series router.

Examples

The following sample output from the **debug pxf tbridge** command shows that the Bridge Group Virtual Interface (BVI) 100 has been removed from the Software Mac-address Filter (SMF) table:

```
Router# debug pxf tbridge

*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Deleting BVI entry 100 from
SMF table.
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: BVI 100 ICM programming
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Successfully removed SMF entry
 for bvi 100
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Deleting BVI entry 100 from
SMF table.
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: BVI 100 ICM programming
*Feb  8 18:39:04.710: rpmxf_tbridge_add_remove_bvi_from_smf: Successfully removed SMF entry
 for bvi 100
*Feb  8 18:39:05.178: %SYS-5-CONFIG_I: Configured from console by vty0
(CROI_MASTER_000A004B)
*Feb  8 18:39:06.710: %LINK-5-CHANGED: Interface BVI100, changed state to administratively
 down
*Feb  8 18:39:07.710:%LINEPROTO-5-UPDOWN: Line protocol on Interface BVI100, changed state
 to down
```

The following sample output from the **debug pxf tbridge** command shows that BVI is configured and that the SMF entry has been updated:

```
Router# debug pxf tbridge

*Feb  8 18:39:16.398:
Note: A random mac address of 0000.0ceb.c0f8 has been chosen for BVI in bridge group  100
since there is no mac address associated with the selected interface.
*Feb  8 18:39:16.398: Ensure that this address is unique.
```

```
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: SMF update for Switch1.1: BVI 100 Mac Address
 0000.0ceb.c0f8
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: BVI 100 ICM programming
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: Successfully updated SMF entry for bvi 100
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: SMF update for Switch1.1:
BVI 100 Mac Address 0000.0ceb.c0f8
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: BVI 100 ICM programming
*Feb  8 18:39:16.398: rpmxf_tbridge_smf_update: Successfully updated SMF entry for bvi 100
*Feb  8 18:39:16.886: %SYS-5-CONFIG_I: Configured from console by vty0
(CROI_MASTER_000A004B)
*Feb  8 18:39:18.394: %LINK-3-UPDOWN: Interface BVI100, changed state to up
*Feb  8 18:39:19.394: %LINEPROTO-5-UPDOWN: Line protocol on Interface BVI100, changed state
 to up
```

Related Commands

Command	Description
show pxf cpu statistics	Displays PXF CPU statistics for a configured router.
show pxf cpu subblock	Displays PXF CPU subblocks for a bridged subinterface.
show pxf cpu tbridge	Displays PXF CPU statistics for transparent bridging.
show pxf statistics	Displays chassis-wide, summary PXF statistics.

debug qbm through debug rudpv1

debug qbm

To display debugging output for quality of service (QoS) bandwidth manager (QBM) options, use the **debug qbm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qbm {**api** | **events**}
no debug qbm {**api** | **events**}

Syntax Description

api	Displays information about QBM client requests and notifications. See the "Usage Guidelines" section for additional information.
events	Displays information about QBM pool events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRC	This command was introduced.
Cisco IOS XE Release 2.6	This command was integrated into Cisco IOS XE Release 2.6.

Usage Guidelines

Use the **debug qbm** command to troubleshoot QBM behavior.

Examples of client requests are when a client creates or destroys a bandwidth pool and when a client attempts to admit bandwidth into a pool. An example of a notification is when a client's previously admitted bandwidth gets preempted from a pool.

Examples

The following example shows how to enable the **debug qbm api**command:

```
Router# debug qbm api
QBM client requests and notifications debugging is on
```

The following example show how to enable the **debug qbm events**command:

```
Router# debug qbm events
QBM pool events debugging is on
```

The following example shows how to verify that QBM debugging is enabled:

```
Router# show debug
QoS Bandwidth Manager:
  QBM client requests and notifications debugging is on
  QBM pool events debugging is on
```

Related Commands

Command	Description
show qbm client	Displays registered QBM clients.
show qbm pool	Displays allocated QBM pools and associated objects.

debug qos dsmib error

To display Quality of Service (QoS) DiffServ MIB errors, use the **debug qos dsmib error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qos dsmib error

no debug qos dsmib error

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.3(1)T	This command was introduced.
XE 3.8S	This command was modified. Support was added for the ASR 1000 Series Routers.

Usage Guidelines

To enable DiffServ MIB support for Qo) policy maps, you use the **qos diffservmib** command. For complete debug information on QoS DiffServ MIB related errors and events, you can enable additional debugging messages using the **debug qos dsmib event** and the **debug qos dsmib stats** commands.

Examples

```
Device# debug qos dsmib error
QoS dsmib error debugging is on
```

Related Commands

Command	Description
debug qos dsmib event	Enables debugging of QoS DiffServ MIB events.
debug qos dsmib stats	Displays QoS DiffServ MIB Statistics.
qos diffservmib	Enables DiffServ MIB support for QoS policy maps.

debug qos dsmib event

To enable debugging of Quality of Service (QoS) DiffServ MIB events, use the **debug qos dsmib event** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

no debug qos dsmib event
debug qos dsmib event

Syntax Description

This command has no arguments or keywords

Command Modes

Privileged EXEC

Command History

Release	Modification
15.3(1)T	This command was introduced.
XE 3.8S	This command was modified. Support was added for the ASR 1000 Series Routers.

Usage Guidelines

To enable DiffServ MIB support for Qo) policy maps, you use the **qos diffservmib** command. For complete debug information on QoS DiffServ MIB related errors and events, you can enable additional debugging messages using the **debug qos dsmib error** and the **debug qos dsmib stats** commands.

Examples

```
Device# debug qos dsmib event
QoS dsmib event debugging is on
```

Related Commands

Command	Description
debug qos dsmib error	Displays QoS DiffServ MIB errors.
debug qos dsmib stats	Displays QoS DiffServ MIB Statistics.
qos diffservmib	Enables DiffServ MIB support for QoS policy maps.

debug qos dsmib stats

To display Quality of Service (QoS) DiffServ MIB statistics, use the **debug qos dsmib error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qos dsmib stats

no debug qos dsmib stats

Syntax Description

This command has no arguments or keywords

Command Modes

Privileged EXEC

Command History

Release	Modification
15.3(1)T	This command was introduced.
XE 3.8S	This command was modified. Support was added for the ASR 1000 Series Routers.

Usage Guidelines

To enable DiffServ MIB support for Qo) policy maps, you use the **qos diffservmib** command. For complete debug information on QoS DiffServ MIB related errors and events, you can enable additional debugging messages using the **debug qos dsmib error** and the **debug qos dsmib events** commands.

Examples

```
Device# debug qos dsmib stats
QoS dsmib stats debugging is on
```

Related Commands

Command	Description
debug qos dsmib errors	Displays QoS DiffServ MIB errors.
debug qos dsmib event	Enables debugging of QoS DiffServ MIB events.
qos diffservmib	Enables DiffServ MIB support for QoS policy maps.

debug qllc error

To display quality link line control (QLLC) errors, use the **debug qllc error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qllc error
no debug qllc error

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command helps you track down errors in the QLLC interactions with X.25 networks. Use the **debug qllc error** command in conjunction with the **debug x25 all** command to see the connection. The data shown by this command only flows through the router on the X.25 connection. Some forms of this command can generate a substantial amount of output and network traffic.

Examples

The following is sample output from the **debug qllc error** command:

```
Router# debug qllc error

%QLLC-3-GENERRMSG: qllc_close - bad qllc pointer Caller 00407116 Caller 00400BD2
QLLC 4000.1111.0002: NO X.25 connection. Discarding XID and calling out
```

The following line indicates that the QLLC connection was closed:

```
%QLLC-3-GENERRMSG: qllc_close - bad qllc pointer Caller 00407116 Caller 00400BD2
```

The following line shows the virtual MAC address of the failed connection:

```
QLLC 4000.1111.0002: NO X.25 connection. Discarding XID and calling out
```

debug qllc event

To enable debugging of quality link line control (QLLC) events, use the **debug qllc event**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qllc event
no debug qllc event

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug qllc event** command to display primitives that might affect the state of a QLLC connection. An example of these events is the allocation of a QLLC structure for a logical channel indicator when an X.25 call has been accepted with the QLLC call user data. Other examples are the receipt and transmission of LAN explorer and exchange identification (XID) frames.

Examples

The following is sample output from the **debug qllc event** command:

```
Router# debug qllc event
QLLC: allocating new qllc lci 9
QLLC: tx POLLING TEST, da 4001.3745.1088, sa 4000.1111.0001
QLLC: rx explorer response, da 4000.1111.0001, sa c001.3745.1088, rif 08B0.1A91.1901.A040
QLLC: gen NULL XID, da c001.3745.1088, sa 4000.1111.0001, rif 0830.1A91.1901.A040, dsap 4,
 ssap 4
QLLC: rx XID response, da 4000.1111.0001, sa c001.3745.1088, rif 08B0.1A91.1901.A040
```

The following line indicates that a new QLLC data structure has been allocated:

```
QLLC: allocating new qllc lci 9
```

The following lines show transmission and receipt of LAN explorer or test frames:

```
QLLC: tx POLLING TEST, da 4001.3745.1088, sa 4000.1111.0001
QLLC: rx explorer response, da 4000.1111.0001, sa c001.3745.1088, rif 08B0.1A91.1901.A040
```

The following lines show XID events:

```
QLLC: gen NULL XID, da c001.3745.1088, sa 4000.1111.0001, rif 0830.1A91.1901.A040, dsap 4,
 ssap 4
QLLC: rx XID response, da 4000.1111.0001, sa c001.3745.1088, rif 08B0.1A91.1901.A040
```

debug qllc packet

To display quality link line control (QLLC) events and QLLC data packets, use the **debug qllc packet** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qllc packet
no debug qllc packet

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command helps you to track down errors in the QLLC interactions with X.25 networks. The data shown by this command only flows through the router on the X25 connection. Use the **debug qllc packet** command in conjunction with the **debug x25 all**command to see the connection and the data that flows through the router.

Examples

The following is sample output from the **debug qllc packet** command:

```
Router# debug qllc packet

14:38:05: Serial2/5 QLLC I: Data Packet.-RSP    9 bytes.
14:38:07: Serial2/6 QLLC I: Data Packet.-RSP 112 bytes.
14:38:07: Serial2/6 QLLC O: Data Packet. 128 bytes.
14:38:08: Serial2/6 QLLC I: Data Packet.-RSP    9 bytes.
14:38:08: Serial2/6 QLLC I: Data Packet.-RSP 112 bytes.
14:38:08: Serial2/6 QLLC O: Data Packet. 128 bytes.
14:38:08: Serial2/6 QLLC I: Data Packet.-RSP    9 bytes.
14:38:12: Serial2/5 QLLC I: Data Packet.-RSP 112 bytes.
14:38:12: Serial2/5 QLLC O: Data Packet. 128 bytes.
```

The following lines indicate that a packet was received on the interfaces:

```
14:38:05: Serial2/5 QLLC I: Data Packet.-RSP    9 bytes.
14:38:07: Serial2/6 QLLC I: Data Packet.-RSP 112 bytes.
```

The following lines show that a packet was sent on the interfaces:

```
14:38:07: Serial2/6 QLLC O: Data Packet. 128 bytes.
14:38:12: Serial2/5 QLLC O: Data Packet. 128 bytes.
```

debug qllc state

To enable debugging of quality link line control (QLLC) events, use the **debug qllc state**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qllc state
no debug qllc state

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

Use the **debug qllc state** command to show when the state of a QLLC connection has changed. The typical QLLC connection goes from states ADM to SETUP to NORMAL. The NORMAL state indicates that a QLLC connection exists and is ready for data transfer.

Examples

The following is sample output from the **debug qllc state** command:

```
Router# debug qllc state
Serial2 QLLC O: QSM-CMD
Serial2: X25 O D1 DATA (5) Q 8 lci 9 PS 4 PR 3
QLLC: state ADM -> SETUP
Serial2: X25 I D1 RR (3) 8 lci 9 PR 5
Serial2: X25 I D1 DATA (5) Q 8 lci 9 PS 3 PR 5
Serial2 QLLC I: QUA-RSPQLLC: addr 00, ctl 73
QLLC: qsetupstate: recvd qua rsp
QLLC: state SETUP -> NORMAL
```

The following line indicates that a QLLC connection attempt is changing state from ADM to SETUP:

```
QLLC: state ADM -> SETUP
```

The following line indicates that a QLLC connection attempt is changing state from SETUP to NORMAL:

```
QLLC: state SETUP -> NORMAL
```

debug qllc timer

To display quality link line control (QLLC) timer events, use the **debug qllc timer** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qllc timer
no debug qllc timer

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The QLLC process periodically cycles and checks status of itself and its partner. If the partner is not found in the desired state, an LAPB primitive command is re-sent until the partner is in the desired state or the timer expires.

Examples

The following is sample output from the **debug qllc timer** command:

```
Router# debug qllc timer

14:27:24: Qllc timer lci 257, state ADM retry count 0 Caller 00407116 Caller 00400BD2
14:27:34: Qllc timer lci 257, state NORMAL retry count 0
14:27:44: Qllc timer lci 257, state NORMAL retry count 1
14:27:54: Qllc timer lci 257, state NORMAL retry count 1
```

The following line of output shows the state of a QLLC partner on a given X.25 logical channel identifier:

```
14:27:24: Qllc timer lci 257, state ADM retry count 0 Caller 00407116 Caller 00400BD2
```

Other messages are informational and appear every ten seconds.

debug qllc x25

To display X.25 packets that affect a quality link line control (QLLC) connection, use the **debugqllcx25** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug qllc x25
no debug qllc x25

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

This command is helpful to track down errors in the QLLC interactions with X.25 networks. Use the **debugqllcx25** command in conjunction with the **debugx25events**or**debugx25all**commands to see the X.25 events between the router and its partner.

Examples

The following is sample output from the **debugqllcx25** command:

```
Router# debug qllc x25

15:07:23: QLLC X25 notify lci 257 event 1
15:07:23: QLLC X25 notify lci 257 event 5
15:07:34: QLLC X25 notify lci 257 event 3 Caller 00407116 Caller 00400BD2
15:07:35: QLLC X25 notify lci 257 event 4
```

The following table describes the significant fields shown in the display.

Table 81: debug qllc x25 Field Descriptions

Field	Description
15:07:23	Displays the time of day.
QLLC X25 notify 257	Indicates that this is a QLLC X25 message.
event <*n*>	Indicates the type of event, *n*. Values for *n*can be as follows: • 1--Circuit is cleared • 2--Circuit has been reset • 3--Circuit is connected • 4--Circuit congestion has cleared • 5--Circuit has been deleted

debug qos accounting

To enable debugging for Quality of Service (QoS) accounting, use the **debug qos accounting** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug qos accounting {**error** | **event** | **ha**}

no debug qos accounting {**error** | **event** | **ha**}

Syntax Description

error	Enables QoS accounting error debugging.
event	Enables QoS accounting event debugging.
ha	Enables QoS accounting high availability debugging.

Command Default

QoS accounting debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced.
Cisco IOS XE Release 3.5S	This command was modified. The **error**, **event**, and **ha** keywords were added.

Usage Guidelines

When QoS policy accounting is enabled on the router, you can use the **debug qos accounting** command to display debugging and troubleshooting information.

Examples

The following example shows how to enable QoS accounting error debugging:

```
Router# debug qos accounting error
QoS accounting error debugging is on
```

The following shows how to disable QoS accounting error debugging:

```
Router# no debug qos accounting error
QoS accounting error debugging is off
```

The following is sample output from the **debug qos accounting ha** command:

```
Router# debug qos accounting ha
*Nov 14 11:12:40.315: PAC CCM HA: [12] add handle: 42000001
*Nov 14 11:12:40.315: PAC CCM HA: found COA cluster handle: 3
*Nov 14 11:12:40.315: PAC CCM HA: [12] set dyn sess required: 42000001 3 1
*Nov 14 11:12:40.315: PAC CCM HA: found COA cluster handle: 3
*Nov 14 11:12:40.315: PAC CCM HA: [12] aaa create flow: 42000001 3 0
*Nov 14 11:12:40.315: PAC CCM HA: found COA cluster handle: 3
*Nov 14 11:12:40.315: PAC CCM HA: [12] dyn sess ready: 42000001 3 1
*Nov 14 11:12:40.315: PAC CCM HA: [12] session sets to periodic updates
*Nov 14 11:12:40.316: PAC CCM HA: [12] get dynsess sync info: items 1, length 58 NAS#
```

```
*Nov 14 11:12:40.316: PAC CCM HA: [12] add all dynsess sync data - max 58
*Nov 14 11:12:40.316: PAC CCM HA: Pulling latest statistics from c3pl beforesync
*Nov 14 11:12:40.316: PAC CCM HA: Collecting HA stats from 2 instances
*Nov 14 11:12:40.316: PAC CCM HA: Collecting HA stats dir input  bytes 0 packets 0
*Nov 14 11:12:40.316: PAC CCM HA: Collecting HA stats dir output  bytes 0 packets 0
*Nov 14 11:12:40.316: PAC CCM HA: xmit xform message type 1
*Nov 14 11:12:40.316: PAC CCM HA: [12] added 58 of dynsess sync CCM data, max 58
```

The following is sample output from the **debug qos accounting event** command:

```
Router# debug qos accounting event
*Nov 14 11:10:33.654: pac: Same group-list mapping is entered
*Nov 14 11:10:33.654: pac: Existing group-list mapping with turbo-service><_GRP default
*Nov 14 11:10:33.656: %SYS-5-CONFIG_I: Configured from console by tty64
*Nov 14 11:10:33.660: pac: event=CLASS_ADD if_info=2A99BC9DB0 cid=0 dir=0 AAA uid=12
*Nov 14 11:10:33.660: pac: Enabling accounting on a class cid: 0 global-parent: [-1 -1 -1
-1] dir: 0
*Nov 14 11:10:33.660: pac: Inserting session 12 into wavl tree
*Nov 14 11:10:33.660: pac: Creating context for group
*Nov 14 11:10:33.660: pac: Added first instance to AAA id: 0xC, group: turbo-service><_GRP
*Nov 14 11:10:33.660: pac: Setting coa_push_mode for context 0x2A99CED228
*Nov 14 11:10:33.660: pac: Updating initial stats dir input bytes 0 packets 0
*Nov 14 11:10:33.661: pac: Username inherited for AAA flow Id
*Nov 14 11:10:33.661: pac: Successfully allocated flow hdl 0x2A000001, id 1 for AAA id 0xC
*Nov 14 11:10:33.661: pac: CoA progressing, WAIT_FOR_COA_ACK, aaa_id 0xC
*Nov 14 11:10:33.662: pac: event=CLASS_ADD if_info=2A99BC9D28 cid=0 dir=1 AAA uid=12
*Nov 14 11:10:33.662: pac: Enabling accounting on a class cid: 0 global-parent: [-1 -1 -1
-1] dir: 1
*Nov 14 11:10:33.662: pac: Adding instance to AAA id: 0xC, group: turbo-service><_GRP
*Nov 14 11:10:33.662: pac: Setting coa_push_mode for context 0x2A99CED228
*Nov 14 11:10:33.662: pac: Updating initial stats dir output bytes 0 packets 0
*Nov 14 11:10:33.663: pac: Preparing to send service start 0xC 0x2A000001 NAS#
*Nov 14 11:10:33.663: pac: Adding session and service static attributes
*Nov 14 11:10:33.663: pac: Service name Nturbo-service() returned with group
turbo-service><_GRP
*Nov 14 11:10:33.663: pac: Configuration: template
*Nov 14 11:10:33.663: pac: Sending Start ...
*Nov 14 11:10:33.663: peruser_acct_callback: Transmitted group in WAIT_FOR_COA_ACK, context
 0x2A99CED228
```

Related Commands

Command	Description
debug qos ha	Enables debugging for QoS high availability information on the networking device.
debug radius	Enables debugging for RADIUS configuration.

debug qos ha

To debug quality of service (QoS) information on the networking device, use the **debug qos ha**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug qos ha [**detail**]
no debug qos ha [**detail**]

Syntax Description

detail	(Optional) Displays detailed debug messages related to specified QoS information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use to determine that QoS in running properly on your networking device.

Examples

The following example enables QoS debugging:

```
Router# debug qos ha
```

debug radius

To enable debugging for Remote Authentication Dial-In User Service (RADIUS) configuration, use the **debug radius** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug radius [{**accounting** | **authentication** | **brief** | **elog** | **failover** | **retransmit** | **verbose**}]
no debug radius [{**accounting** | **authentication** | **brief** | **elog** | **failover** | **retransmit** | **verbose**}]

Syntax Description

accounting	(Optional) Enables debugging of RADIUS accounting collection.
authentication	(Optional) Enables debugging of RADIUS authentication packets.
brief	(Optional) Displays abbreviated debug output.
elog	(Optional) Enables RADIUS event logging.
failover	(Optional) Enables debugging of packets sent upon failover.
retransmit	(Optional) Enables retransmission of packets.
verbose	(Optional) Displays detailed debug output.

Command Default

RADIUS event logging and debugging output in ASCII format are enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.2(1)T	This command was introduced.
12.0(2)T	The **brief** keyword was added. The default output format became ASCII from hexadecimal.
12.2(11)T	The **verbose** keyword was added.
12.3(2)T	The **elog** keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

RADIUS is a distributed security system that secures networks against unauthorized access. Cisco supports RADIUS under the authentication, authorization, and accounting (AAA) security system. When RADIUS is used on the router, you can use the **debug radius** command to display debugging and troubleshooting information in ASCII format. Use the **debug radius brief** command for abbreviated output displaying client/server interaction and minimum packet information. Only the input and output transactions are recorded. Use the **debug radius verbose** command to include non-essential RADIUS debugs.

Examples

The following is sample output from the **debug radius** command:

```
Router# debug radius
Radius protocol debugging is on
Radius packet hex dump debugging is off
Router# show debug
00:19:20: RADIUS/ENCODE(00000015):Orig. component type = AUTH_PROXY
00:19:20: RADIUS(00000015): Config NAS IP: 0.0.0.0
00:19:20: RADIUS/ENCODE(00000015): acct_session_id: 21
00:19:20: RADIUS(00000015): sending
00:19:20: RADIUS/ENCODE: Best Local IP-Address 33.0.0.2 for Radius-Server 33.2.0.1
00:19:20: RADIUS(00000015): Send Access-Request to 33.2.0.1:1645 id 1645/21, len 159
00:19:20: RADIUS:  authenticator 2D 03 E5 A6 A5 30 1A 32 - F2 C5 EE E2 AC 5E 5D 22
00:19:20: RADIUS:  User-Name           [1]   11  "authproxy"
00:19:20: RADIUS:  User-Password       [2]   18  *
00:19:20: RADIUS:  Service-Type        [6]   6   Outbound                  [5]
00:19:20: RADIUS:  Message-Authenticato[80]  18
00:19:20: RADIUS:   85 EF E8 43 03 88 58 63 78 D2 7B E7 26 61 D3 3C          [ CXcx{&a<]
00:19:20: RADIUS:  Vendor, Cisco       [26]  49
00:19:20: RADIUS:   Cisco AVpair      [1]   43  "audit-session-id=0D00000200000013001112FD"
00:19:20: RADIUS:  NAS-Port-Type       [61]  6   Ethernet                  [15]
00:19:20: RADIUS:  NAS-Port            [5]   6   16480
00:19:20: RADIUS:  NAS-Port-Id         [87]  19  "FastEthernet1/0/3"
00:19:20: RADIUS:  NAS-IP-Address      [4]   6   33.0.0.2
00:19:20: RADIUS(00000015): Started 5 sec timeout
00:19:20: RADIUS: Received from id 1645/21 33.2.0.1:1645, Access-Accept, len 313
00:19:20: RADIUS:  authenticator E6 6E 1D 64 5A 15 FD AE - C9 60 C0 68 F5 10 E9 B7
00:19:20: RADIUS:  Filter-Id           [11]  8
00:19:20: RADIUS:   31 30 30 2E 69 6E           [ 100.in]
00:19:20: RADIUS:  Vendor, Cisco       [26]  19
00:19:20: RADIUS:   Cisco AVpair       [1]   13  "priv-lvl=15"
00:19:20: RADIUS:  Termination-Action  [29]  6   1
00:19:20: RADIUS:  Vendor, Cisco       [26]  45
00:19:20: RADIUS:   Cisco AVpair       [1]   39  "supplicant-name=Port-description test"
00:19:20: RADIUS:  Vendor, Cisco       [26]  38
00:19:20: RADIUS:   Cisco AVpair       [1]   32  "security-group-tag=2468-C0FFEE"
00:19:20: RADIUS:  Vendor, Cisco       [26]  33
00:19:20: RADIUS:   Cisco AVpair       [1]   27  "supplicant-group=engineer"
00:19:20: RADIUS:  Vendor, Cisco       [26]  36
00:19:20: RADIUS:   Cisco AVpair       [1]   30  "supplicant-group=idf_testing"
00:19:20: RADIUS:  Vendor, Cisco       [26]  28
00:19:20: RADIUS:   Cisco AVpair       [1]   22  "authz-directive=open"
00:19:20: RADIUS:  Vendor, Cisco       [26]  32
00:19:20: RADIUS:   Cisco AVpair       [1]   26  "supplicant-group=group-9"
00:19:20: RADIUS:  Class               [25]  30
00:19:20: RADIUS:   43 41 43 53 3A 63 2F 61 37 31 38 38 61 2F 32 31  [CACS:c/a7188a/21]
00:19:20: RADIUS:   30 30 30 30 30 32 2F 31 36 34 38 30      [ 000002/16480]
00:19:20: RADIUS:  Message-Authenticato[80]  18
00:19:20: RADIUS:   24 13 29 95 A1 5E 9F D3 CB ED 78 F1 F6 62 2B E3            [ $)^xb+]
00:19:20: RADIUS(00000015): Received from id 1645/21
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "supplicant-group" - IGNORE
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "supplicant-group" - IGNORE
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "authz-directive" - IGNORE
00:19:20: RADIUS/DECODE: parse unknown cisco vsa "supplicant-group" - IGNORE
00:19:20: RADIUS/ENCODE(00000015):Orig. component type = AUTH_PROXY
00:19:20: RADIUS(00000015): Config NAS IP: 0.0.0.0
00:19:20: RADIUS(00000015): sending
00:19:20: RADIUS/ENCODE: Best Local IP-Address 33.0.0.2 for Radius-Server 33.2.0.1
00:19:20: RADIUS(00000015): Send Accounting-Request to 33.2.0.1:1646 id 1646/1, len 204
00:19:20: RADIUS:  authenticator A7 6B A0 94 F4 63 30 51 - 8A CE 8C F4 8A 8E 0B CC
00:19:20: RADIUS:  Acct-Session-Id     [44]  10  "00000015"
00:19:20: RADIUS:  Calling-Station-Id  [31]  10  "13.1.0.1"
00:19:20: RADIUS:  Vendor, Cisco       [26]  49
00:19:20: RADIUS:   Cisco AVpair      [1]   43  "audit-session-id=0D00000200000013001112FD"
```

The following is sample output from the **debug radius brief** command:

```
Router# debug radius brief
Radius protocol debugging is on
Radius packet hex dump debugging is off
Radius protocol in brief format debugging is on
00:05:21: RADIUS: Initial Transmit ISDN 0:D:23 id 6 10.0.0.1:1824, Accounting-Request, len
 358
00:05:21: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085274206
00:05:26: RADIUS: Retransmit id 6
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No valid server found. Trying any viable server
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No response for id 7
00:05:31: RADIUS: Initial Transmit ISDN 0:D:23 id 8 10.0.0.0:1823, Access-Request, len 171
00:05:36: RADIUS: Retransmit id 8
00:05:36: RADIUS: Received from id 8 1.7.157.1:1823, Access-Accept, len 115
00:05:47: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085274206, call lasted
 26 seconds
00:05:47: RADIUS: Initial Transmit ISDN 0:D:23 id 9 10.0.0.1:1824, Accounting-Request, len
 775
00:05:47: RADIUS: Received from id 9 1.7.157.1:1824, Accounting-response, len 20
```

The following example shows how to enable debugging of RADIUS accounting collection:

```
Router# debug radius accounting
Radius protocol debugging is on
Radius protocol brief debugging is off
Radius protocol verbose debugging is off
Radius packet hex dump debugging is off
Radius packet protocol (authentication) debugging is off
Radius packet protocol (accounting) debugging is on
Radius packet retransmission debugging is off
Radius server fail-over debugging is off
Radius elog debugging is off
```

Related Commands

Command	Description
debug aaa accounting	Displays information on accountable events as they occur.
debug aaa authentication	Displays information on AAA/TACACS+ authentication.

debug radius local-server

To control the display of debug messages for the local authentication server, use the **debug radius local-server**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug radius local-server {**client** | **error** | **packets**}
no debug radius local-server {**client** | **error** | **packets**}

Syntax Description

client	Displays error messages about failed client authentications.
error	Displays error messages about the local authentication server.
packets	Displays the content of the RADIUS packets that are sent and received.

Command Default

No default behavior or values

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(11)JA	This command was introduced on Cisco Aironet Access Point 1200 and Cisco Aironet Access Point 1100.
12.3(11)T	This command was implemented on the following platforms: Cisco 2600XM, Cisco 2691, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3700, and Cisco 3800 series routers.
12.4(2)T	This command was integrated into Cisco IOS Release 12.4(2)T.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to control the display of debug messages for the local authentication server.

Examples

The following command shows how to display messages regarding failed client authentication:

```
Router# debug radius local-server
 client
```

Related Commands

Command	Description
clear radius local-server	Clears the statistics display or unblocks a user.
show radius local-server statistics	Displays statistics for a local network access server.
ssid	Specifies up to 20 SSIDs to be used by a user group.
user	Authorizes a user to authenticate using the local authentication server.

Command	Description
vlan	Specifies a VLAN to be used by members of a user group.

debug radius-proxy

To display debugging messages for Intelligent Services Gateway (ISG) RADIUS proxy functionality, use the **debug radius-proxy**command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug radius-proxy {**events** | **errors**}
no debug radius-proxy {**events** | **errors**}

Syntax Description

events	Displays debug messages related to ISG RADIUS proxy events.
errors	Displays debug messages related to ISG RADIUS proxy errors.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(31)SB2	This command was introduced.

Usage Guidelines

See the following caution before using **debug** commands.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, only use **debug** commands to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use debug commands during periods of lower network flows and fewer users.

Examples

The following example shows output for the **debug radius-proxy** command with the **events** keyword:

```
Router# debug radius-proxy events
*Nov  7 07:53:11.411: RP-EVENT: Parse Request: Username = 12345679@cisco
*Nov  7 07:53:11.411: RP-EVENT: Parse Request: Caller ID = 12345679@cisco
*Nov  7 07:53:11.411: RP-EVENT: Parse Request: NAS id = localhost
*Nov  7 07:53:11.411: RP-EVENT: Found matching context for user Caller ID:12345679@cisco
Name:aa
*Nov  7 07:53:11.411: RP-EVENT: Received event client Access-Request in state activated
*Nov  7 07:53:11.411: RP-EVENT: User Caller ID:12345679@cisco  Name:12 re-authenticating
*Nov  7 07:53:11.411: RP-EVENT: Forwarding Request to method list (handle=1979711512)
*Nov  7 07:53:11.411: RP-EVENT: Sending request to server group EAP
*Nov  7 07:53:11.411: RP-EVENT: State changed activated --> wait for Access-Response
```

debug rai

To enable debugging for Resource Allocation Indication (RAI), use the **debug rai** command in privileged EXEC mode. To disable debugging for RAI, use the **no** form of this command.

debug rai
no debug rai

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(2)T	This command was introduced.

Usage Guidelines

You can use the **debug rai** command along with the **debug ccsip all** command to get the complete debugging information for RAI.

Examples

The following example shows how to enable resource allocation debugging:

```
Router# debug rai
Resource Availability debugging is on
*Dec 16 05:50:34.863: //1/rai_new_resource_index:New index created 1
*Dec 16 05:50:34.863: //1/rai_main:- event code:7
*Dec 16 05:50:34.863: //1/rai_process_new_rsc_group:New Resource Index created
*Dec 16 05:50:34.907: //1/rai_set_resource_info_config:Resource type 0
*Dec 16 05:50:34.907: //1/rai_set_resource_info_config:New system resource created 0x4961A38C
*Dec 16 05:50:34.907: //1/rai_set_resource_info_config:Resource New Config Event passed
*Dec 16 05:50:34.907: //1/rai_set_resource_info_config:Resource Type CPU Subtype 1-min-avg
 Low watermark 30High watermark 50
*Dec 16 05:50:34.907: //1/rai_main:- event code:4
```

Related Commands

Command	Description
periodic-report interval	Configures periodic reporting parameters for gateway resource entities.
rai target	Configures the SIP RAI mechanism.
resource (voice)	Configures parameters for monitoring resources, use the resource command in voice-class configuration mode.
show voice class resource-group	Displays the resource group configuration information for a specific resource group or all resource groups.
voice class resource-group	Enters voice-class configuration mode and assigns an identification tag number for a resource group.

debug ras

To display the types and addressing of Registration, Admission and Status (RAS) messages sent and received, use the **debug ras** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ras
no debug ras

Syntax Description

This command has arguments or keywords.

Command Default

This command is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(6)NA2	This command was introduced.
12.2(2)XB1	This command was implemented on the Cisco AS5850 universal access router.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.

Usage Guidelines

Use the **debug ras** command to display the types and addressing of RAS messages sent and received. The debug output lists the message type using mnemonics defined in International Telecommunications Union-Telecommunication (ITU-T) specification H.225.

Examples

In the following output, gateway GW13.cisco.com sends a RAS registration request (RRQ) message to gatekeeper GK15.cisco.com at IP address 10.9.53.15. GW13.cisco.com then receives a registration confirmation (RCF) message from the gatekeeper. If there is no response, it could mean that the gatekeeper is offline or improperly addressed. If you receive a reject (RRJ) message, it could mean that the gatekeeper is unable to handle another gateway or that the registration information is incorrect.

```
Router# debug ras

*Mar 13 19:53:34.231:      RASlib::ras_sendto:msg length 105 from
                            10.9.53.13:8658 to 10.9.53.15:1719
*Mar 13 19:53:34.231:      RASLib::RASSendRRQ:RRQ (seq# 36939) sent
                            to 10.9.53.15
*Mar 13 19:53:34.247:      RASLib::RASRecvData:successfully rcvd
                            message of length 105 from 10.9.53.15:1719
*Mar 13 19:53:34.251:      RASLib::RASRecvData:RCF (seq# 36939) rcvd
                            from [10.9.53.15:1719] on sock [0x6168356C]
```

debug redundancy application group asymmetric-routing

To log debug information for an asymmetric routing redundancy application group, use the **debug redundancy application group asymmetric-routing** command in privileged EXEC mode. To disable the debug log, use the **no** form of this command.

debug redundancy application group asymmetric-routing [{**error** | **peer** | **tunnel**}]
no debug redundancy application group asymmetric-routing [{**error** | **peer** | **tunnel**}]

Syntax Description

error	(Optional) Specifies the asymmetric routing redundancy group errors.
peer	(Optional) Specifies the asymmetric routing redundancy group peer events.
tunnel	(Optional) Specifies the asymmetric routing redundancy tunnel events.

Command Default

Debugging of asymmetric routing redundancy group is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.2(3)T	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group asymmetric-routing peer** command:

```
On standby :
*Mar  6 20:57:25.996: RG-AR-PEER: RG AR:group 1 start negotiation timer
*Mar  6 20:57:26.006: RG-AR-PEER: RG AR:group 1 stop negotiation timer
*Mar  6 20:57:26.006: RG-AR-PEER: RG AR:group 1 transport negotiated

On Active:
*Mar  6 20:57:26.006: RG-AR-PEER: RG AR:group 1 stop negotiation timer
*Mar  6 20:57:26.006: RG-AR-PEER: RG AR:group 1 transport negotiated
```

The following is sample output from the **debug redundancy application group asymmetric-routing tunnel** command:

```
On standby:
*Mar  6 20:52:25.886: RG-AR-TUNNEL: encap packet(len 114) for redirection, orig pak encsize
 14
*Mar  6 20:52:25.886: RG-AR-TUNNEL: packet(len 132) redirected successfully for feature (1)
 from rii (1000) group (1)

On Active:
Case 1: CEF enabled
*Mar  6 20:52:25.887: RG-AR-TUNNEL: packet(len 146) received in CEF path
*Mar  6 20:52:25.887: RG-AR-TUNNEL: packet received for group (1) rii (1000) forwarded using
 parent idb Ethernet1/3
Case 2: CEF disabled)
*Mar  6 20:54:45.449: RG-AR-TUNNEL: packet(len 100) received for group (1) rii (1000
Ethernet1/3) from standby received in process path
```

Related Commands

Command	Description
redundancy asymmetric-routing enable	Establishes an asymmetric flow diversion tunnel for each redundancy group.

debug redundancy application group config

To display the redundancy application group configuration, use the **debug redundancy application group config**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group config {**all** | **error** | **event** | **func**}
no debug redundancy application group config {**all** | **error** | **event** | **func**}

Syntax Description

all	Displays debug information about the configuration.
error	Displays information about the redundancy group's configuration errors.
event	Displays information about the redundancy group's configuration.
func	Displays information about the redundancy group's configuration functions entered.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group config all**command:

```
Router# debug redundancy application group config all

RG config all debugging is on
```

Related Commands

Command	Description
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy application group transport information.
debug redundancy application group vp	Displays the redundancy application group VP information.

debug redundancy application group faults

To display the redundancy application group faults, use the **debug redundancy application group faults**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group faults {**all** | **error** | **event** | **fault** | **func**}
no debug redundancy application group faults {**all** | **error** | **event** | **fault** | **func**}

Syntax Description

all	Displays fault information of a redundancy group.
error	Displays error information of a redundancy groups.
event	Displays event information of a redundancy group.
fault	Displays fault events information of a redundancy group.
func	Displays fault functions information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group faults error**command:

```
Router# debug redundancy application group faults error
RG Faults error debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy application group configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy group application group transport information.
debug redundancy application group vp	Displays the redundancy group application group VP information.

debug redundancy application group media

To display the redundancy application group media information, use the **debug redundancy application group media**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group media {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}
no debug redundancy application group media {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}

Syntax Description

all	Displays media information of a redundancy group.
error	Displays media error information of a redundancy group.
event	Displays media events information of a redundancy group.
nbr	Displays media neighbor (nbr) information of a redundancy group.
packet	Displays media packets information of a redundancy group.
rx	Displays the incoming packets information.
tx	Displays the outgoing packets information.
timer	Displays information about redundancy group media timer events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group media timer**command:

```
Router# debug redundancy application group media timer
RG Media timer debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group protocol	Displays the redundancy group application group protocol information.
debug redundancy application group rii	Displays the redundancy group application group RII information.
debug redundancy application group transport	Displays the redundancy group application group transport information.

Command	Description
debug redundancy application group vp	Displays the redundancy application group VP information.

debug redundancy application group protocol

To display the redundancy application group protocol information, use the **debug redundancy application group protocol**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group protocol {**all** | **detail** | **error** | **event** | **media** | **peer**}
no debug redundancy application group protocol {**all** | **detail** | **error** | **event** | **media** | **peer**}

Syntax Description

all	Displays protocol information of a redundancy group.
detail	Displays event details of a redundancy group.
error	Displays protocol error information of a redundancy group.
event	Displays protocol events information of a redundancy group.
media	Displays protocol media events information of a redundancy group.
peer	Displays protocol peer information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group protocol peer**command:

```
Router# debug redundancy application group protocol peer

RG Protocol peer debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy application group transport information.

Command	Description
debug redundancy application group vp	Displays the redundancy application group VP information.

debug redundancy application group rii

To display the redundancy application group redundancy interface identifier (RII) information, use the **debug redundancy application group rii**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group rii {**error** | **event**}
no debug redundancy application group rii {**error** | **event**}

Syntax Description

error	Displays RII error information of a redundancy group.
event	Displays RII event information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group rii event**command:

```
Router# debug redundancy application group rii event

RG RII events debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy group application group protocol information.
debug redundancy application group vp	Displays the redundancy group application group VP information.

debug redundancy application group transport

To display the redundancy application group transport information, use the **debug redundancy application group transport**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group transport {**db** | **error** | **event** | **packet** | **timer** | **trace**}
no debug redundancy application group transport {**db** | **error** | **event** | **packet** | **timer** | **trace**}

Syntax Description

db	Displays transport information of a redundancy group.
error	Displays transport error information of a redundancy group.
event	Displays transport event information of a redundancy group.
packet	Displays transport packet information of a redundancy group.
timer	Displays transport timer information of a redundancy group.
trace	Displays transport trace information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group transport trace**command:

```
Router# debug redundancy application group transport trace
RG Transport trace debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.

debug redundancy application group vp

To display the redundancy application group virtual platform (VP) information, use the **debug redundancy application group vp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group vp {**error** | **event**}
no debug redundancy application group vp {**error** | **event**}

Syntax Description

error	Displays VP error information of a redundancy group.
event	Displays VP event information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group vp event**command:

```
Router# debug redundancy application group vp event

RG VP events debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy application group transport information.

debug redundancy (RP)

To enable the display of events for troubleshooting dual Route Processors (RPs), use the **debug redundancy**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug redundancy {**ehsa** | **errors** | **fsm** | **kpa** | **msg** | **progression** | **status** | **timer**}
no debug redundancy {**ehsa** | **errors** | **fsm** | **kpa** | **msg** | **progression** | **status** | **timer**}

Syntax Description

ehsa	Displays redundancy facility (RF) enhanced high system availability (EHSA) information.
errors	Displays RF errors.
fsm	Displays RF feasible successor metrics (FSM) events.
kpa	Displays RF keepalive events.
msg	Displays RF messaging events.
progression	Displays RF progression events.
status	Displays RF status events.
timer	Displays RF timer events.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.3(6)AA	This command was introduced.
12.0(15)ST	This command was introduced on Cisco 10000 series Internet routers.
12.0(22)S	This command was introduced on Cisco 7500, 10000, and 12000 series Internet routers.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S on Cisco 7500 series routers.
12.2(20)S	Support was added for the Cisco 7304 router. The Cisco 7500 series router is not supported in Cisco IOS Release 12.2(20)S.
12.2(28)SB	Support for this command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example enables debugging information for RF keepalive events:

```
Router# debug redundancy kpa
```

debug redundancy application group config

To display the redundancy application group configuration, use the **debug redundancy application group config**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group config {**all** | **error** | **event** | **func**}
no debug redundancy application group config {**all** | **error** | **event** | **func**}

Syntax Description

all	Displays debug information about the configuration.
error	Displays information about the redundancy group's configuration errors.
event	Displays information about the redundancy group's configuration.
func	Displays information about the redundancy group's configuration functions entered.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group config all**command:

```
Router# debug redundancy application group config all

RG config all debugging is on
```

Related Commands

Command	Description
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy application group transport information.
debug redundancy application group vp	Displays the redundancy application group VP information.

debug redundancy application group faults

To display the redundancy application group faults, use the **debug redundancy application group faults**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group faults {**all** | **error** | **event** | **fault** | **func**}
no debug redundancy application group faults {**all** | **error** | **event** | **fault** | **func**}

Syntax Description

all	Displays fault information of a redundancy group.
error	Displays error information of a redundancy groups.
event	Displays event information of a redundancy group.
fault	Displays fault events information of a redundancy group.
func	Displays fault functions information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group faults error**command:

```
Router# debug redundancy application group faults error
RG Faults error debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy application group configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy group application group transport information.
debug redundancy application group vp	Displays the redundancy group application group VP information.

debug redundancy application group media

To display the redundancy application group media information, use the **debug redundancy application group media**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group media {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}
no debug redundancy application group media {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}

Syntax Description

all	Displays media information of a redundancy group.
error	Displays media error information of a redundancy group.
event	Displays media events information of a redundancy group.
nbr	Displays media neighbor (nbr) information of a redundancy group.
packet	Displays media packets information of a redundancy group.
rx	Displays the incoming packets information.
tx	Displays the outgoing packets information.
timer	Displays information about redundancy group media timer events.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group media timer**command:

```
Router# debug redundancy application group media timer
RG Media timer debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group protocol	Displays the redundancy group application group protocol information.
debug redundancy application group rii	Displays the redundancy group application group RII information.
debug redundancy application group transport	Displays the redundancy group application group transport information.

Command	Description
debug redundancy application group vp	Displays the redundancy application group VP information.

debug redundancy application group protocol

To display the redundancy application group protocol information, use the **debug redundancy application group protocol**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group protocol {**all** | **detail** | **error** | **event** | **media** | **peer**}
no debug redundancy application group protocol {**all** | **detail** | **error** | **event** | **media** | **peer**}

Syntax Description

all	Displays protocol information of a redundancy group.
detail	Displays event details of a redundancy group.
error	Displays protocol error information of a redundancy group.
event	Displays protocol events information of a redundancy group.
media	Displays protocol media events information of a redundancy group.
peer	Displays protocol peer information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group protocol peer**command:

```
Router# debug redundancy application group protocol peer

RG Protocol peer debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy application group transport information.

Command	Description
debug redundancy application group vp	Displays the redundancy application group VP information.

debug redundancy application group rii

To display the redundancy application group redundancy interface identifier (RII) information, use the **debug redundancy application group rii**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group rii {**error** | **event**}
no debug redundancy application group rii {**error** | **event**}

Syntax Description

error	Displays RII error information of a redundancy group.
event	Displays RII event information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group rii event**command:

```
Router# debug redundancy application group rii event

RG RII events debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy group application group protocol information.
debug redundancy application group vp	Displays the redundancy group application group VP information.

debug redundancy application group transport

To display the redundancy application group transport information, use the **debug redundancy application group transport**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group transport {**db** | **error** | **event** | **packet** | **timer** | **trace**}
no debug redundancy application group transport {**db** | **error** | **event** | **packet** | **timer** | **trace**}

Syntax Description

db	Displays transport information of a redundancy group.
error	Displays transport error information of a redundancy group.
event	Displays transport event information of a redundancy group.
packet	Displays transport packet information of a redundancy group.
timer	Displays transport timer information of a redundancy group.
trace	Displays transport trace information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group transport trace**command:

```
Router# debug redundancy application group transport trace
RG Transport trace debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.

debug redundancy application group vp

To display the redundancy application group virtual platform (VP) information, use the **debug redundancy application group vp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy application group vp {**error** | **event**}
no debug redundancy application group vp {**error** | **event**}

Syntax Description

error	Displays VP error information of a redundancy group.
event	Displays VP event information of a redundancy group.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.1S	This command was introduced.

Examples

The following is sample output from the **debug redundancy application group vp event**command:

```
Router# debug redundancy application group vp event

RG VP events debugging is on
```

Related Commands

Command	Description
debug redundancy application group config	Displays the redundancy group application configuration.
debug redundancy application group media	Displays the redundancy application group media information.
debug redundancy application group protocol	Displays the redundancy application group protocol information.
debug redundancy application group rii	Displays the redundancy application group RII information.
debug redundancy application group transport	Displays the redundancy application group transport information.

debug redundancy as5850

To enable specific redundancy-related debug options, use the **debug redundancy as5850**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug redundancy as5850 {**fsm** | **lines** | **master** | **mode** | **rf-client**}
no debug redundancy as5850

Syntax Description

fsm	Finite-state-machine events.
lines	Hardware lines.
master	Master (active rather than standby) route-switch-controller (RSC).
mode	RSC's mode: classic-split or handover-split.
rf-client	Redundancy-related client-application information.

Command Default

This command is disabled

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XB1	This command was introduced.
12.2(11)T	This command was integrated into Cisco IOS Release 12.2(11)T.

Usage Guidelines

Use the master form of the command to view redundancy-related debug entries. All debug entries continue to be logged even if you do not specify an option here, and you can always use the **show redundancy debug-log** command to view them.

Examples

The output from this command consists of event announcements that can be used by authorized troubleshooting personnel.

Related Commands

Command	Description
show redundancy debug-log	Displays up to 256 debug entries.

debug registry

To turn on the debugging output for registry events or errors when Cisco IOS Software Modularity software is running, use the **debug registry** command in privileged EXEC mode. To turn off debugging output, use the **no** form of this command or the **undebug**command.

debug registry {**events** | **errors**} [{*process-namepid*}]
no debug registry {**events** | **errors**} [{*process-namepid*}]

Syntax Description

events	Displays debugging messages about registry event messages.
errors	Displays debugging messages about registry error messages.
process-name	(Optional) Process name.
pid	(Optional) Process ID. Number in the range from 1 to 4294967295.

Command Modes

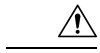
Privileged EXEC

Command History

Release	Modification
12.2(18)SXF4	This command was introduced to support Software Modularity images.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **debug registry** command to troubleshoot Software Modularity registry operations.



Caution Use any debugging command with caution because the volume of generated output can slow or stop the router operations. We recommend that this command be used only under the supervision of a Cisco engineer.

Examples

The following example turns on debugging messages for Software Modularity registry events for the TCP process:

```
Router# debug registry events tcp.proc
Debug registry events debugging is on
```

The following example turns on debugging messages for Software Modularity registry errors:

```
Router# debug registry errors
Debug registry errors debugging is on
```

debug resource policy notification

To trace the Embedded Resource Manager (ERM) notification activities for resources using the ERM feature, use the **debug resource policy notification** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug resource policy notification [**owner** *resource-owner-name*]
no debug resource policy notification [**owner** *resource-owner-name*]

Syntax Description

owner *resource-owner-name*	(Optional) Specifies the name of the resource owner (RO).

Command Default

Disabled

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following example shows different instances of the **debug resource policy notification**command:

```
Router# debug resource policy notification
Enabled notif. debugs on all owners
```

When a threshold is violated, the following messages are displayed:

```
*Mar  3 09:50:44.081: Owner: 'memory' initiated a notification:
*Mar  3 09:50:44.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major memory
 threshold
Pool: Processor Used: 42932864 Threshold :42932860
*Mar  3 09:50:46.081: Notification from Owner: 'memory' is dispatched for User: 'usrr1'
(ID: 0x10000B9)
*Mar  3 09:50:46.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major memory
 threshold
Pool: Processor Used: 42932864 Threshold :42932860
Router# no debug resource manager notification

Disabled notif. debugs on all owners
Router# debug resource manager notification owner cpu

Enabled notif. debugs on owner 'cpu'
Router# no debug resource manager notification owner cpu

Disabled notif. debugs on owner 'cpu'
Router# debug resource manager notification owner memory
```

```
Enabled notif. debugs on owner 'memory'
Router# no debug resource manager notification owner memory

Disabled notif. debugs on owner 'memory'
Router# debug resource manager notification owner Buffer

Enabled notif. debugs on owner 'Buffer'
Router# no debug resource manager notification owner Buffer

Disabled notif. debugs on owner 'Buffer'
Router# no debug resource manager notification owner Buffer
Disabled notif. debugs on owner 'Buffer'
```

Related Commands

Command	Description
debug resource policy registration	Displays the resource policy registration debug information for the ERM resources.

debug resource policy registration

To trace the Embedded Resource Manager (ERM) registration activities for resources using the ERM feature, use the **debug resource policy registration**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug resource policy registration
no debug resource policy registration

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Examples

The following example shows output from the **debug resource policy registration**command:

```
Router# debug resource policy registration

Registrations debugging is on
```

When a Resource User (RU) is created, the following message is displayed:

```
*Mar  3 09:35:58.304: resource_user_register: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID: 0x0
name: usrr1
```

When an RU is deleted, the following message is displayed:

```
*Mar  3 09:41:09.500: resource_user_unregister: RU: ruID: 0x10000B8, rutID: 0x1, rg_ID: 0x0
 name: usrr1
```

Related Commands

Command	Description
debug resource policy notification	Displays the resource policy notification debug information for the ERM resources.

debug resource-pool

To see and trace resource pool management activity, use the **debugresource-pool** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug resource-pool
no debug resource-pool

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(4)XI	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Enter the **debugresource-pool** command to see and trace resource pool management activity. The following table describes the resource pooling states.

Table 82: Resource Pooling States

State	Description
RM_IDLE	No call activity.
RM_RES_AUTHOR	Call waiting for authorization, message sent to authentication, authorization, and accounting (AAA).
RM_RES_ALLOCATING	Call authorized, resource-grp-mgr allocating.
RM_RES_ALLOCATED	Resource allocated, connection acknowledgment sent to signalling state. Call should get connected and become active.
RM_AUTH_REQ_IDLE	Signalling module disconnected call while in RM_RES_AUTHOR. Waiting for authorization response from AAA.
RM_RES_REQ_IDLE	Signalling module disconnected call while in RM_RES_ALLOCATING. Waiting for resource allocation response from resource-group manager.
RM_DNIS_AUTHOR	An intermediate state before proceeding with Route Processor Module (RPM) authorization.
RM_DNIS_AUTH_SUCCEEDED	Dialed number identification service (DNIS) authorization succeeded.
RM_DNIS_RES_ALLOCATED	DNIS resource allocated.
RM_DNIS_AUTH_REQ_IDLE	DNIS authorization request idle.

State	Description
RM_DNIS_AUTHOR_FAIL	DNIS authorization failed.
RM_DNIS_RES_ALLOC_SUCCESS	DNIS resource allocation succeeded.
RM_DNIS_RES_ALLOC_FAIL	DNIS resource allocation failed.
RM_DNIS_RPM_REQUEST	DNIS resource pool management requested.

You can use the resource pool state to isolate problems. For example, if a call fails authorization in the RM_RES_AUTHOR state, investigate further with AAA authorization debugs to determine whether the problem lies in the resource-pool manager, AAA, or dispatcher.

Examples

The following example shows different instances where you can use the **debugresource-pool**command:

```
Router# debug resource-pool
RM general debugging is on
Router# show debug
General OS:
  AAA Authorization debugging is on
Resource Pool:
  resource-pool general debugging is on
Router #
Router #ping 21.1.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 21.1.1.10, timeout is 2 seconds:
*Jan  8 00:10:30.358: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:1
*Jan  8 00:10:30.358: RM: event incoming call
/* An incoming call is received by RM */
*Jan  8 00:10:30.358: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST
DS0:0:0:0:1
/* Receives an event notifying to proceed with RPM authorization while
in DNIS authorization state */
*Jan  8 00:10:30.358: RM:RPM event incoming call
*Jan  8 00:10:30.358: RPM profile cp1 found
/* A customer profile "cp1" is found  matching for the incoming call, in
the local database */
*Jan  8 00:10:30.358: RM state:RM_RPM_RES_AUTHOR
event:RM_RPM_RES_AUTHOR_SUCCESS DS0:0:0:0:1
/* Resource authorization success event received while in  resource
authorization state*/
*Jan  8 00:10:30.358: Allocated resource from res_group isdn1
*Jan  8 00:10:30.358: RM:RPM profile "cp1", allocated resource "isdn1"
successfully
*Jan  8 00:10:30.358: RM state:RM_RPM_RES_ALLOCATING
event:RM_RPM_RES_ALLOC_SUCCESS DS0:0:0:0:1
/* Resource allocation sucess event received while attempting to
allocate a resource */
*Jan  8 00:10:30.358: Se0:1 AAA/ACCT/RM: doing resource-allocated
(local) (nothing to do)
*Jan  8 00:10:30.366: %LINK-3-UPDOWN: Interface Serial0:1, changed state
to up
*Jan  8 00:10:30.370: %LINK-3-UPDOWN: Interface Serial0:1, changed state
to down
*Jan  8 00:10:30.570: Se0:1 AAA/ACCT/RM: doing resource-update (local)
cp1 (nothing to do)
*Jan  8 00:10:30.578: %LINK-3-UPDOWN: I.nterface Serial0:0, changed
```

```
state to up
*Jan  8 00:10:30.582: %DIALER-6-BIND: Interface Serial0:0 bound to
profile Dialer0...
Success rate is 0 percent (0/5)
Router #
*Jan  8 00:10:36.662: %ISDN-6-CONNECT: Interface Serial0:0 is now
connected to 71017
*Jan  8 00:10:52.990: %DIALER-6-UNBIND: Interface Serial0:0 unbound from
profile Dialer0
*Jan  8 00:10:52.990: %ISDN-6-DISCONNECT: Interface Serial0:0
disconnected from 71017 , call lasted 22 seconds
*Jan  8 00:10:53.206: %LINK-3-UPDOWN: Interface Serial0:0, changed state
to down
*Jan  8 00:10:53.206: %ISDN-6-DISCONNECT: Interface Serial0:1
disconnected from unknown , call lasted 22 seconds
*Jan  8 00:10:53.626: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON
DS0:0:0:0:1
/* Received  Disconnect event from signalling stack  for a call which
has a resource allocated. */
*Jan  8 00:10:53.626: RM:RPM event call drop
/* RM processing the disconnect event */
*Jan  8 00:10:53.626: Deallocated resource from res_group isdn1
*Jan  8 00:10:53.626: RM state:RM_RPM_DISCONNECTING
event:RM_RPM_DISC_ACK DS0:0:0:0:1
/* An intermediate state while the DISCONNECT event is being processed
by external servers, before RM goes back into IDLE state.
*/
```

The following table describes the significant fields shown in the display.

Table 83: debug resource-pool Field Descriptions

Field	Description
RM state:RM_IDLE	Resource manager state that displays no active calls.
RM state:RM_RES_AUTHOR	Resource authorization state.
RES_AUTHOR_SUCCESS DS0: shelf:slot:port:channel	Actual physical resource that is used
Allocated resource from res_group	Physical resource group that accepts the call.
RM profile <x>, allocated resource <x>	Specific customer profile and resource group names used to accept the call.
RM state: RM_RES_ALLOCATING	Resource manager state that unifies a call with a physical resource.

debug rif

To display information on entries entering and leaving the routing information field (RIF) cache, use the **debugrif** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rif
no debug rif

Syntax Description

This command has no arguments or keywords.

Command Modes

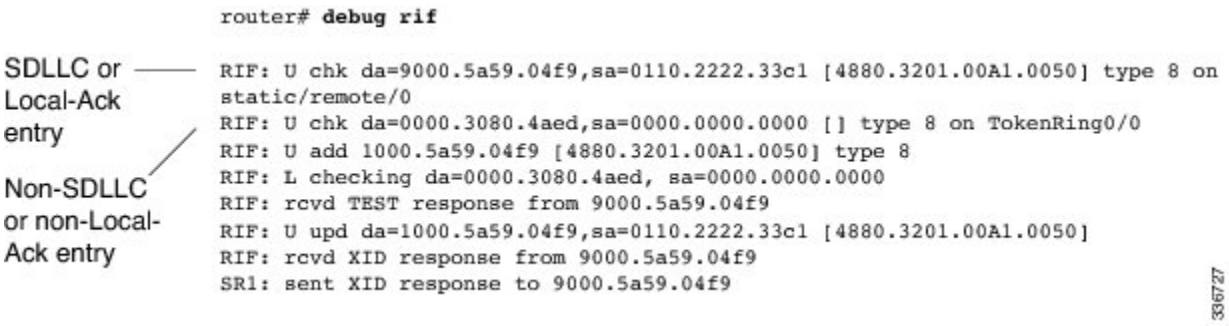
Privileged EXEC

Usage Guidelines

In order to use the **debugrif** command to display traffic source-routed through an interface, fast switching of source route bridging (SRB) frames must first be disabled with the **nosource-bridgeroute-cache** interface configuration command.

Examples

The following is sample output from the **debugrif** command:



The first line of output is an example of a RIF entry for an interface configured for SDLC Logical Link Control (SDLLC) or Local-Ack. The following table describes significant fields shown in the display.

Table 84: debug rif Field Descriptions

Field	Description
RIF:	This message describes RIF debugging output.
U chk	Update checking. The entry is being updated; the timer is set to zero (0).
da=9000.5a59.04f9	Destination MAC address.
sa=0110.2222.33c1	Source MAC address. This field contains values of zero (0000.0000.0000) in a non-SDLLC or non-Local-Ack entry.
[4880.3201.00A1.0050]	RIF string. This field is blank (null RIF) in a non-SDLLC or non-Local-Ack entry.

Field	Description
type 8	Possible values follow: • 0--Null entry • 1--This entry was learned from a particular Token Ring port (interface) • 2--Statically configured • 4--Statically configured for a remote interface • 8--This entry is to be aged • 16--This entry (which has been learned from a remote interface) is to be aged • 32--This entry is not to be aged • 64--This interface is to be used by LAN Network Manager (and is not to be aged)
on static/remote/0	This route was learned from a real Token Ring port, in contrast to a virtual ring.

The following line of output is an example of a RIF entry for an interface that is not configured for SDLLC or Local-Ack:

```
RIF: U chk da=0000.3080.4aed,sa=0000.0000.0000 [] type 8 on TokenRing0/0
```

Notice that the source address contains only zero values (0000.0000.0000), and that the RIF string is null ([ ]). The last element in the entry indicates that this route was learned from a virtual ring, rather than a real Token Ring port.

The following line shows that a new entry has been added to the RIF cache:

```
RIF: U add 1000.5a59.04f9 [4880.3201.00A1.0050] type 8
```

The following line shows that a RIF cache lookup operation has taken place:

```
RIF: L checking da=0000.3080.4aed, sa=0000.0000.0000
```

The following line shows that a TEST response from address 9000.5a59.04f9 was inserted into the RIF cache:

```
RIF: rcvd TEST response from 9000.5a59.04f9
```

The following line shows that the RIF entry for this route has been found and updated:

```
RIF: U upd da=1000.5a59.04f9,sa=0110.2222.33c1 [4880.3201.00A1.0050]
```

The following line shows that an XID response from this address was inserted into the RIF cache:

```
RIF: rcvd XID response from 9000.5a59.04f9
```

The following line shows that the router sent an XID response to this address:

```
SR1: sent XID response to 9000.5a59.04f9
```

The following table explains the other possible lines of **debugrif** command output.

Table 85: Additional debug rif Field Descriptions

Field	Description
RIF: L Sending XID for <*address*>	Router/bridge wanted to send a packet to *address* but did not find it in the RIF cache. It sent an XID explorer packet to determine which RIF it should use. The attempted packet is dropped.
RIF: L No buffer for XID to <*address*>	Similar to the previous description; however, a buffer in which to build the XID packet could not be obtained.
RIF: U remote rif too small <*rif*>	Packet's RIF was too short to be valid.
RIF: U rej <*address*> too big <*rif*>	Packet's RIF exceeded the maximum size allowed and was rejected. The maximum size is 18 bytes.
RIF: U upd interface <*address*>	RIF entry for this router/bridge's interface has been updated.
RIF: U ign <*address*> interface update	RIF entry that would have updated an interface corresponding to one of this router's interfaces.
RIF: U add <*address*><*rif*>	RIF entry for *address* has been added to the RIF cache.
RIF: U no memory to add rif for <*address*>	No memory to add a RIF entry for *address*.
RIF: removing rif entry for <*address*,*typecode*>	RIF entry for *address* has been forcibly removed.
RIF: flushed <*address*>	RIF entry for *address* has been removed because of a RIF cache flush.
RIF: expired <*address*>	RIF entry for *address* has been aged out of the RIF cache.

Related Commands

Command	Description
debug list	Filters debugging information on a per-interface or per-access list basis.

debug route-map ipc

To display a summary of the one-way Inter-process Communications (IPC) messages set from the route processor (RP) to the Versatile Interface Processor (VIP) about NetFlow policy routing when distributed Cisco Express Forwarding (dCEF) is enabled, use the **debug route-map ipc**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug route-map ipc commanddebug route-map ipc
no debug route-map ipc commanddebug route-map ipc

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(3)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command is especially helpful for policy routing with dCEF switching.

This command displays a summary of one-way IPC messages from the RP to the VIP about NetFlow policy routing. If you execute this command on the RP, the messages are shown as “Sent.” If you execute this command on the VIP console, the IPC messages are shown as “Received.”

Examples

The following is sample output from the **debug route-map ipc**command executed at the RP:

```
Router# debug route-map ipc
Routemap related IPC debugging is on

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip cef distributed
Router(config)#^Z
Router#
RM-IPC: Clean routemap config in slot 0
RM-IPC: Sent clean-all-routemaps; len 12
RM-IPC: Download all policy-routing related routemap config to slot 0
RM-IPC: Sent add routemap test(seq:10); n_len 5; len 17
RM-IPC: Sent add acl 1 of routemap test(seq:10); len 21
RM-IPC: Sent add min 10 max 300 of routemap test(seq:10); len 24
RM-IPC: Sent add preced 1 of routemap test(seq:10); len 17
RM-IPC: Sent add tos 4 of routemap test(seq:10); len 17
RM-IPC: Sent add nexthop 50.0.0.8 of routemap test(seq:10); len 20
RM-IPC: Sent add default nexthop 50.0.0.9 of routemap test(seq:10); len 20
RM-IPC: Sent add interface Ethernet0/0/3(5) of routemap test(seq:10); len 20
RM-IPC: Sent add default interface Ethernet0/0/2(4) of routemap test(seq:10); len 20
```

The following is sample output from the **debug route-map ipc**command executed at the VIP:

```
VIP-Slot0# debug route-map ipc
Routemap related IPC debugging is on
```

```
VIP-Slot0#
RM-IPC: Rcvd clean-all-routemaps; len 12
RM-IPC: Rcvd add routemap test(seq:10); n_len 5; len 17
RM-IPC: Rcvd add acl 1 of routemap test(seq:10); len 21
RM-IPC: Rcvd add min 10 max 300 of routemap test(seq:10); len 24
RM-IPC: Rcvd add preced 1 of routemap test(seq:10); len 17
RM-IPC: Rcvd add tos 4 of routemap test(seq:10); len 17
RP-IPC: Rcvd add nexthop 50.0.0.8 of routemap test(seq:10); len 20
RP-IPC: Rcvd add default nexthop 50.0.0.9 of routemap test(seq:10); len 20
RM-IPC: Rcvd add interface Ethernet0/3 of routemap tes; len 20
RM-IPC: Rcvd add default interface Ethernet0/2 of routemap test(seq:10); len 20
```

debug rpms-proc preauth

To enable diagnostic reporting of preauthentication information, use the **debugrpms-procpreauth** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rpms-proc preauth {**all** | **h323** | **sip**}
no debug rpms-proc preauth {**all** | **h323** | **sip**}

Syntax Description

all	Provides information for all calls.
h323	Provides information for H.323 calls.
sip	Provides information for Session Initiation Protocol (SIP) calls.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(11)T	This command was introduced.

Examples

The following example shows debugging output for two calls. The first is a leg 3 SIP call, and the second is a leg 3 H.323 call:

```
Router# debug rpms-proc preauth all
All RPMS Process preauth tracing is enabled
Feb 10 14:00:07.236: Entering rpms_proc_print_preauth_req
Feb 10 14:00:07.236: Request = 0
Feb 10 14:00:07.236: Preauth id = 8
Feb 10 14:00:07.236: EndPt Type  = 1
Feb 10 14:00:07.236: EndPt = 192.168.80.70
Feb 10 14:00:07.236: Resource Service = 1
Feb 10 14:00:07.236: Call_origin = answer
Feb 10 14:00:07.236: Call_type = voip
Feb 10 14:00:07.236: Calling_num = 2220001
Feb 10 14:00:07.236: Called_num = 1120001
Feb 10 14:00:07.236: Protocol  = 1
Feb 10 14:00:07.236:rpms_proc_create_node:Created node with preauth_id = 8
Feb 10 14:00:07.236:rpms_proc_send_aaa_req:uid got is 19
Feb 10 14:00:07.240:rpms_proc_preauth_response:Context is for preauth_id 8, aaa_uid 19
Feb 10 14:00:07.240:rpms_proc_preauth_response:Deleting Tree node for preauth id 8 uid 19
Feb 10 14:00:07.284: Entering rpms_proc_print_preauth_req
Feb 10 14:00:07.284: Request = 0
Feb 10 14:00:07.284: Preauth id = 9
Feb 10 14:00:07.284: EndPt Type  = 1
Feb 10 14:00:07.284: EndPt = 192.168.81.102
Feb 10 14:00:07.284: Resource Service = 1
Feb 10 14:00:07.284: Call_origin = answer
Feb 10 14:00:07.284: Call_type = voip
Feb 10 14:00:07.284: Calling_num = 2210001
Feb 10 14:00:07.284: Called_num = 1#1110001
Feb 10 14:00:07.284: Protocol  = 0
Feb 10 14:00:07.288:rpms_proc_create_node:Created node with preauth_id = 9
```

```
Feb 10 14:00:07.288:rpms_proc_send_aaa_req:uid got is 21
Feb 10 14:00:07.300:rpms_proc_preauth_response:Context is for preauth_id 9, aaa_uid 21
Feb 10 14:00:07.300:rpms_proc_preauth_response:Deleting Tree node for preauth id 9 uid 21
```

The following example shows the output for a single leg 3 H.323 call:

```
Router# debug rpms-proc preauth h323

RPMS Process H323 preauth tracing is enabled
Feb 10 14:04:57.867: Entering rpms_proc_print_preauth_req
Feb 10 14:04:57.867: Request = 0
Feb 10 14:04:57.867: Preauth id = 10
Feb 10 14:04:57.867: EndPt Type  = 1
Feb 10 14:04:57.867: EndPt = 192.168.81.102
Feb 10 14:04:57.867: Resource Service = 1
Feb 10 14:04:57.867: Call_origin = answer
Feb 10 14:04:57.867: Call_type = voip
Feb 10 14:04:57.867: Calling_num = 2210001
Feb 10 14:04:57.867: Called_num = 1#1110001
Feb 10 14:04:57.867: Protocol  = 0
Feb 10 14:04:57.867:rpms_proc_create_node:Created node with preauth_id = 10
Feb 10 14:04:57.867:rpms_proc_send_aaa_req:uid got is 25
Feb 10 14:04:57.875:rpms_proc_preauth_response:Context is for preauth_id 10, aaa_uid 25
Feb 10 14:04:57.875:rpms_proc_preauth_response:Deleting Tree node for preauth id 10 uid 25
```

The following example shows output for a single leg 3 SIP call:

```
Router# debug rpms-proc preauth sip
RPMS Process SIP preauth tracing is enabled
Feb 10 14:08:02.880: Entering rpms_proc_print_preauth_req
Feb 10 14:08:02.880: Request = 0
Feb 10 14:08:02.880: Preauth id = 11
Feb 10 14:08:02.880: EndPt Type  = 1
Feb 10 14:08:02.880: EndPt = 192.168.80.70
Feb 10 14:08:02.880: Resource Service = 1
Feb 10 14:08:02.880: Call_origin = answer
Feb 10 14:08:02.880: Call_type = voip
Feb 10 14:08:02.880: Calling_num = 2220001
Feb 10 14:08:02.880: Called_num = 1120001
Feb 10 14:08:02.880: Protocol  = 1
Feb 10 14:08:02.880:rpms_proc_create_node:Created node with preauth_id = 11
Feb 10 14:08:02.880:rpms_proc_send_aaa_req:uid got is 28
Feb 10 14:08:02.888:rpms_proc_preauth_response:Context is for preauth_id 11, aaa_uid 28
Feb 10 14:08:02.888:rpms_proc_preauth_response:Deleting Tree node for preauth id 11 uid 28
```

The following table describes the significant fields shown in the display.

Table 86: debug rpms-proc preauth Field Descriptions

Field	Description
Request	Request Type--0 for preauthentication, 1 for disconnect.
Preauth id	Identifier for the preauthentication request.
EndPt Type	Call Origin End Point Type--1 for IP address, 2 for Interzone ClearToken (IZCT) value.
EndPt	Call Origin End Point Value--An IP address or IZCT value.
Resource Service	Resource Service Type--1 for Reservation, 2 for Query.

Field	Description
Call_origin	Answer.
Call_type	Voice over IP (VoIP).
Calling_num	Calling party number (calling line identification, or CLID).
Called_num	Called party number (dialed number identification service, or DNIS).
Protocol	0 for H.323, 1 for SIP.
function reports	Various identifiers and status reports for executed functions.

debug rtpspi all

To debug all Routing Table Protocol (RTP) security parameter index (SPI) errors, sessions, and in/out functions, use the **debug rtpspi all** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtpspi all
no debug rtpspi all

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XK	This command was introduced on the Cisco MC3810 and Cisco 3600 series routers (except the Cisco 3620).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Caution Be careful when you use this command because it can result in console flooding and reduced voice quality.

Examples

The following example shows a debug trace for RTP SPI errors, sessions, and in/out functions on a gateway:

```
Router# debug rtpspi all
RTP SPI Error, Session and function in/out tracings are enabled.
*Mar  1 00:38:59.381:rtpspi_allocate_rtp_port:Entered.
*Mar  1 00:38:59.381:rtpspi_allocate_rtp_port:allocated RTP port 16544
*Mar  1 00:38:59.381:rtpspi_allocate_rtp_port:Success. port = 16544. Leaving.
*Mar  1 00:38:59.381:rtpspi_call_setup_request:entered.
        Call Id = 5, dest = 0.0.0.0;   callInfo:
        final dest flag = 0,
        rtp_session_mode = 0x2,
        local_ip_addrs = 0x5000001,remote_ip_addrs = 0x0,
        local rtp port = 16544, remote rtp port = 0
*Mar  1 00:38:59.381:rtpspi_call_setup_request:spi_info copied for rtpspi_app_data_t.
*Mar  1 00:38:59.385:rtpspi_call_setup_request:leaving
*Mar  1 00:38:59.385:rtpspi_call_setup() entered
*Mar  1 00:38:59.385:rtpspi_initialize_ccb:Entered
*Mar  1 00:38:59.385:rtpspi_initialize_ccb:leaving
*Mar  1 00:38:59.385:rtpspi_call_setup:rtp_session_mode = 0x2
*Mar  1 00:38:59.385:rtpspi_call_setup:mode = CC_CALL_NORMAL.
        destianation number = 0.0.0.0
*Mar  1 00:38:59.385:rtpspi_call_setup:Passed local_ip_addrs=0x5000001
*Mar  1 00:38:59.385:rtpspi_call_setup:Passed local_rtp_port = 16544
*Mar  1 00:38:59.385:rtpspi_call_setup:Saved RTCP Session = 0x1AF57E0
```

```
*Mar  1 00:38:59.385:rtpspi_call_setup:Passed remote rtp port = 0.
*Mar  1 00:38:59.389:rtpspi_start_rtcp_session:entered. rtp session mode=0x2, rem rtp=0,
rem ip=0x0
*Mar  1 00:38:59.389:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x2
*Mar  1 00:38:59.389:rtpspi_start_rtcp_session:Starting RTCP session.
        Local IP addr = 0x5000001, Remote IP addr = 0x0,
        Local RTP port = 16544, Remote RTP port = 0, mode = 0x2
*Mar  1 00:38:59.389:rtpspi_start_rtcp_session:RTP Session creation Success.
*Mar  1 00:38:59.389:rtpspi_call_setup:RTP Session creation Success.
*Mar  1 00:38:59.389:rtpspi_call_setup:calling cc_api_call_connected()
*Mar  1 00:38:59.389:rtpspi_call_setup:Leaving.
*Mar  1 00:38:59.393:rtpspi_bridge:entered. conf id = 1, src i/f = 0x1859E88,
        dest i/f = 0x1964EEC, src call id = 5, dest call id = 4
        call info = 0x1919140, xmit fn = 0xDA7494, tag = 0
*Mar  1 00:38:59.393:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x2
*Mar  1 00:38:59.393:rtpspi_modify_rtcp_session_parameters():xmit fn=0xDA7494,
dstIF=0x1964EEC, dstCallID=4, voip_mode=0x2, rtp_mode=0x2, ssrc_status=0
*Mar  1 00:38:59.393:rtpspi_bridge:Calling cc_api_bridge_done() for 5(0x1AF5400) and 4(0x0).
*Mar  1 00:38:59.393:rtpspi_bridge:leaving.
*Mar  1 00:38:59.397:rtpspi_caps_ind:Entered. vdb = 0x1859E88 call id = 5, srcCallId = 4
*Mar  1 00:38:59.397:rtpspi_caps_ind:caps from VTSP:codec=0x83FB, codec_bytes=0x50,
                   fax rate=0x7F, vad=0x3 modem=0x0
*Mar  1 00:38:59.397:rtpspi_get_rtcp_session_parameters():CURRENT VALUES:
dstIF=0x1964EEC, dstCallID=4, current_seq_num=0x0
*Mar  1 00:38:59.397:rtpspi_get_rtcp_session_parameters():NEW VALUES:
dstIF=0x1964EEC, dstCallID=4, current_seq_num=0x261C
*Mar  1 00:38:59.397:rtpspi_caps_ind:Caps Used:codec=0x1, codec bytes=80,
        fax rate=0x1, vad=0x1, modem=0x1, dtmf_relay=0x1, seq_num_start=0x261D
*Mar  1 00:38:59.397:rtpspi_caps_ind:calling cc_api_caps_ind().
*Mar  1 00:38:59.397:rtpspi_caps_ind:Returning success
*Mar  1 00:38:59.397:rtpspi_caps_ack:Entered. call id = 5, srcCallId = 4
*Mar  1 00:38:59.397:rtpspi_caps_ack:leaving.
*Mar  1 00:38:59.618:rtpspi_call_modify:entered. call-id=5, nominator=0x7, params=0x18DD440
*Mar  1 00:38:59.618:rtpspi_call_modify:leaving
*Mar  1 00:38:59.618:rtpspi_do_call_modify:Entered. call-id = 5
*Mar  1 00:38:59.622:rtpspi_do_call_modify:Remote RTP port changed. New port=16432
*Mar  1 00:38:59.622:rtpspi_do_call_modify:Remote IP addrs changed. New IP addrs=0x6000001
*Mar  1 00:38:59.622:rtpspi_do_call_modify:new mode 2 is the same as the current mode
*Mar  1 00:38:59.622:rtpspi_do_call_modify:Starting new RTCP session.
*Mar  1 00:38:59.622:rtpspi_start_rtcp_session:entered. rtp session mode=0x2, rem rtp=16432,
 rem ip=0x6000001
*Mar  1 00:38:59.622:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x2
*Mar  1 00:38:59.622:rtpspi_start_rtcp_session:Removing old RTCP session.
*Mar  1 00:38:59.622:rtpspi_start_rtcp_session:Starting RTCP session.
        Local IP addr = 0x5000001, Remote IP addr = 0x6000001,
        Local RTP port = 16544, Remote RTP port = 16432, mode = 0x2
*Mar  1 00:38:59.622:rtpspi_start_rtcp_session:RTCP Timer creation Success. (5)*(5000)
*Mar  1 00:38:59.622:rtpspi_start_rtcp_session:RTP Session creation Success.
*Mar  1 00:38:59.622:rtpspi_do_call_modify:RTP Session creation Success.
*Mar  1 00:38:59.622:rtpspi_do_call_modify:Calling cc_api_call_modify(), result=0x0
*Mar  1 00:38:59.626:rtpspi_do_call_modify:success. leaving
*Mar  1 00:39:05.019:rtpspi_call_modify:entered. call-id=5, nominator=0x7, params=0x18DD440
*Mar  1 00:39:05.019:rtpspi_call_modify:leaving
*Mar  1 00:39:05.019:rtpspi_do_call_modify:Entered. call-id = 5
*Mar  1 00:39:05.019:rtpspi_do_call_modify:New remote RTP port = old rtp port = 16432
*Mar  1 00:39:05.019:rtpspi_do_call_modify:New remote IP addrs = old IP addrs = 0x6000001
*Mar  1 00:39:05.019:rtpspi_do_call_modify:Mode changed. new = 3, old = 2
*Mar  1 00:39:05.019:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x3
*Mar  1 00:39:05.023:rtpspi_modify_rtcp_session_parameters():xmit fn=0xDA7494,
dstIF=0x1964EEC, dstCallID=4, voip_mode=0x3, rtp_mode=0x3, ssrc_status=2
*Mar  1 00:39:05.023:rtpspi_do_call_modify:RTCP Timer start.
*Mar  1 00:39:05.023:rtpspi_do_call_modify:Calling cc_api_call_modify(), result=0x0
*Mar  1 00:39:05.023:rtpspi_do_call_modify:success. leaving
*Mar  1 00:40:13.786:rtpspi_bridge_drop:entered. src call-id=5, dest call-id=4, tag=0
```

```
*Mar  1 00:40:13.786:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x3
*Mar  1 00:40:13.786:rtpspi_modify_rtcp_session_parameters():xmit fn=0x0,
dstIF=0x0, dstCallID=0, voip_mode=0x3, rtp_mode=0x3, ssrc_status=2
*Mar  1 00:40:13.786:rtpspi_bridge_drop:leaving
*Mar  1 00:40:13.790:rtpspi_call_disconnect:entered. call-id=5, cause=16, tag=0
*Mar  1 00:40:13.790:rtpspi_call_disconnect:leaving.
*Mar  1 00:40:13.790:rtpspi_do_call_disconnect:Entered. call-id = 5
*Mar  1 00:40:13.790:rtpspi_do_call_disconnect:calling rtpspi_call_cleanup(). call-id=5
*Mar  1 00:40:13.794:rtpspi_call_cleanup:entered. ccb = 0x1AF5400, call-id=5, rtp port =
16544
*Mar  1 00:40:13.794:rtpspi_call_cleanup:releasing ccb cache. RTP port=16544
*Mar  1 00:40:13.794:rtpspi_store_call_history_entry():Entered.
*Mar  1 00:40:13.794:rtpspi_store_call_history_entry():Leaving.
*Mar  1 00:40:13.794:rtpspi_call_cleanup:RTCP Timer Stop.
*Mar  1 00:40:13.794:rtpspi_call_cleanup:deallocating RTP port 16544.
*Mar  1 00:40:13.794:rtpspi_free_rtcp_session:Entered.
*Mar  1 00:40:13.794:rtpspi_free_rtcp_session:Success. Leaving
*Mar  1 00:40:13.794::rtpspi_call_cleanup freeing ccb (0x1AF5400)
*Mar  1 00:40:13.794:rtpspi_call_cleanup:leaving
*Mar  1 00:40:13.794:rtpspi_do_call_disconnect:leaving
```

Related Commands

Command	Description
debug rtpspi errors	Debugs RTP SPI errors.
debug rtpspi inout	Debugs RTP SPI in/out functions.
debug rtpspi send-nse	Triggers the RTP SPI to send a triple redundant NSE.
debug sgcp errors	Debugs SGCP errors.
debug sgcp events	Debugs SGCP events.
debug sgcp packet	Debugs SGCP packets.
debug vtsp send-nse	Sends and debugs a triple redundant NSE from the DSP to a remote gateway.

debug rtpspi errors

To debug Routing Table Protocol (RTP) security parameter index (SPI) errors, use the **debug rtpspi errors**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtpspi errors
no debug rtpspi errors

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XK	This command was introduced on the Cisco MC3810 device and Cisco 3600 series routers (except the Cisco 3620).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Caution Be careful when you use this command because it can result in console flooding and reduced voice quality.

Examples

This example shows a debug trace for RTP SPI errors on two gateways. The following example shows the debug trace on the first gateway:

```
Router# debug rtpspi errors
00:54:13.272:rtpspi_do_call_modify:new mode 2 is the same as the current mode
00:54:18.738:rtpspi_do_call_modify:New remote RTP port = old rtp port = 16452
00:54:18.738:rtpspi_do_call_modify:New remote IP addrs = old IP addrs = 0x6000001
```

The following example shows the debug trace on the second gateway:

```
Router# debug rtpspi errors
00:54:08:rtpspi_process_timers:
00:54:08:rtpspi_process_timers:Timer 0x1A5AF9C expired.
00:54:08:rtpspi_process_timers:Timer expired for callID 0x3
00:54:08:rtpspi_process_timers:
00:54:08:rtpspi_process_timers:Timer 0x1A5AF9C expired.
00:54:08:rtpspi_process_timers:Timer expired for callID 0x3
00:54:08:rtpspi_process_timers:
00:54:08:rtpspi_process_timers:Timer 0x1A5AF9C expired.
00:54:08:rtpspi_process_timers:Timer expired for callID 0x3
00:54:09:rtpspi_process_timers:
00:54:09:rtpspi_process_timers:Timer 0x1A5AFBC expired.
00:54:09:rtpspi_process_timers:Timer expired for callID 0x3
00:54:09:rtpspi_process_timers:
00:54:09:rtpspi_process_timers:Timer 0x1A5B364 expired.
00:54:09:rtpspi_process_timers:Timer expired for callID 0x3
```

Related Commands

Command	Description
debug rtpspi all	Debugs all RTP SPI errors, sessions, and in/out functions.
debug rtpspi inout	Debugs RTP SPI in/out functions.
debug rtpspi send-nse	Triggers the RTP SPI to send a triple redundant NSE.
debug sgcp errors	Debugs SGCP errors.
debug sgcp events	Debugs SGCP events.
debug sgcp packet	Debugs SGCP packets.
debug vtsp send-nse	Sends and debugs a triple redundant NSE from the DSP to a remote gateway.

debug rtpspi inout

To debug Routing Table Protocol (RTP) security parameter index (SPI) in/out functions, use the **debug rtpspi inout**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtpspi inout
no debug rtpspi inout

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XK	This command was introduced on the Cisco MC3810 device and Cisco 3600 series routers (except the Cisco 3620 device).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Caution Be careful when you use this command because it can result in console flooding and reduced voice quality.

Examples

The following example shows a debug trace for RTP SPI in/out functions on a gateway:

```
Router#
debug rtpspi inout
*Mar  1 00:57:24.565:rtpspi_allocate_rtp_port:Entered.
*Mar  1 00:57:24.565:rtpspi_allocate_rtp_port:Success. port = 16520. Leaving.
*Mar  1 00:57:24.565:rtpspi_call_setup_request:entered.
        Call Id = 9, dest = 0.0.0.0;   callInfo:
        final dest flag = 0,
        rtp_session_mode = 0x2,
        local_ip_addrs = 0x5000001,remote_ip_addrs = 0x0,
        local rtp port = 16520, remote rtp port = 0
*Mar  1 00:57:24.565:rtpspi_call_setup_request:spi_info copied for rtpspi_app_data_t.
*Mar  1 00:57:24.565:rtpspi_call_setup_request:leaving
*Mar  1 00:57:24.569:rtpspi_call_setup() entered
*Mar  1 00:57:24.569:rtpspi_initialize_ccb:Entered
*Mar  1 00:57:24.569:rtpspi_initialize_ccb:leaving
*Mar  1 00:57:24.569:rtpspi_start_rtcp_session:entered. rtp session mode=0x2, rem rtp=0,
rem ip=0x0
*Mar  1 00:57:24.569:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x2
*Mar  1 00:57:24.569:rtpspi_call_setup:Leaving.
*Mar  1 00:57:24.573:rtpspi_bridge:entered. conf id = 3, src i/f = 0x1859E88,
        dest i/f = 0x1964EEC, src call id = 9, dest call id = 8
        call info = 0x1919140, xmit fn = 0xDA7494, tag = 0
*Mar  1 00:57:24.573:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x2
*Mar  1 00:57:24.573:rtpspi_bridge:leaving.
*Mar  1 00:57:24.573:rtpspi_caps_ind:Entered. vdb = 0x1859E88 call id = 9, srcCallId = 8
```

```
*Mar  1 00:57:24.577:rtpspi_caps_ind:Returning success
*Mar  1 00:57:24.577:rtpspi_caps_ack:Entered. call id = 9, srcCallId = 8
*Mar  1 00:57:24.577:rtpspi_caps_ack:leaving.
*Mar  1 00:57:24.818:rtpspi_call_modify:entered. call-id=9, nominator=0x7, params=0x18DD440
*Mar  1 00:57:24.818:rtpspi_call_modify:leaving
*Mar  1 00:57:24.818:rtpspi_do_call_modify:Entered. call-id = 9
*Mar  1 00:57:24.818:rtpspi_start_rtcp_session:entered. rtp session mode=0x2, rem rtp=16396,
 rem ip=0x6000001
*Mar  1 00:57:24.822:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x2
*Mar  1 00:57:24.822:rtpspi_do_call_modify:success. leaving
*Mar  1 00:57:30.296:rtpspi_call_modify:entered. call-id=9, nominator=0x7, params=0x18DD440
*Mar  1 00:57:30.296:rtpspi_call_modify:leaving
*Mar  1 00:57:30.300:rtpspi_do_call_modify:Entered. call-id = 9
*Mar  1 00:57:30.300:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x3
*Mar  1 00:57:30.300:rtpspi_do_call_modify:success. leaving
*Mar  1 00:58:39.055:rtpspi_bridge_drop:entered. src call-id=9, dest call-id=8, tag=0
*Mar  1 00:58:39.055:rtpspi_get_rtcp_mode:entered. rtp_mode = 0x3
*Mar  1 00:58:39.055:rtpspi_bridge_drop:leaving
*Mar  1 00:58:39.059:rtpspi_call_disconnect:entered. call-id=9, cause=16, tag=0
*Mar  1 00:58:39.059:rtpspi_call_disconnect:leaving.
*Mar  1 00:58:39.059:rtpspi_do_call_disconnect:Entered. call-id = 9
*Mar  1 00:58:39.059:rtpspi_call_cleanup:entered. ccb = 0x1AF5400, call-id=9, rtp port =
16520
*Mar  1 00:58:39.059:rtpspi_store_call_history_entry():Entered.
*Mar  1 00:58:39.059:rtpspi_store_call_history_entry():Leaving.
*Mar  1 00:58:39.059:rtpspi_free_rtcp_session:Entered.
*Mar  1 00:58:39.059:rtpspi_free_rtcp_session:Success. Leaving
*Mar  1 00:58:39.063:rtpspi_call_cleanup:leaving
*Mar  1 00:58:39.063:rtpspi_do_call_disconnect:leaving
```

Related Commands

Command	Description
debug rtpspi all	Debugs all RTP SPI errors, sessions, and in/out functions.
debug rtpspi errors	Debugs RTP SPI errors.
debug rtpspi send-nse	Triggers the RTP SPI to send a triple redundant NSE.
debug sgcp errors	Debugs SGCP errors.
debug sgcp events	Debugs SGCP events.
debug sgcp packet	Debugs SGCP packets.
debug vtsp send-nse	Sends and debugs a triple redundant NSE from the DSP to a remote gateway.

debug rtpspi send-nse

To trigger the Routing Table Protocol (RTP) security parameter index (SPI) software module to send a triple redundant NSE, use the **debug rtpspi send-nse**command in privileged EXEC mode. To disable this action, use the **no** form of the command.

debug rtpspi send-nse *call-ID NSE-event-ID*
no debug rtpspi send-nse *call-ID NSE-event-ID*

Syntax Description

call-ID	Specifies the call ID of the active call. The valid range is from 0 to 65535.
NSE-event-ID	Specifies the NSE Event ID. The valid range is from 0 to 255.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XK	This command was introduced on the Cisco MC3810 device and Cisco 3600 series routers (except the Cisco 3620 router).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows the RTP SPI software module set to send an NSE:

```
Router# debug rtpspi send-nse
```

Related Commands

Command	Description
debug rtpspi all	Debugs all RTP SPI errors, sessions, and in/out functions.
debug rtpspi errors	Debugs RTP SPI errors.
debug rtpspi inout	Debugs RTP SPI in/out functions.
debug sgcp errors	Debugs SGCP errors.
debug sgcp events	Debugs SGCP events.
debug sgcp packet	Debugs SGCP packets.
debug vtsp send-nse	Sends and debugs a triple redundant NSE from the DSP to a remote gateway.

debug rtpspi session

To debug all Routing Table Protocol (RTP) security parameter index (SPI) sessions, use the **debug rtpspi session**command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug rtpspi session
no debug rtpspi session

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(7)XK	This command was introduced on the Cisco MC3810 device and Cisco 3600 series routers (except the Cisco 3620 router).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows a debug trace for RTP SPI sessions on a gateway:

```
Router# debug rtpspi session
*Mar  1 01:01:51.593:rtpspi_allocate_rtp_port:allocated RTP port 16406
*Mar  1 01:01:51.593:rtpspi_call_setup:rtp_session_mode = 0x2
*Mar  1 01:01:51.593:rtpspi_call_setup:mode = CC_CALL_NORMAL.
       destianation number = 0.0.0.0
*Mar  1 01:01:51.593:rtpspi_call_setup:Passed local_ip_addrs=0x5000001
*Mar  1 01:01:51.593:rtpspi_call_setup:Passed local_rtp_port = 16406
*Mar  1 01:01:51.593:rtpspi_call_setup:Saved RTCP Session = 0x1AFDFBC
*Mar  1 01:01:51.593:rtpspi_call_setup:Passed remote rtp port = 0.
*Mar  1 01:01:51.598:rtpspi_start_rtcp_session:Starting RTCP session.
       Local IP addr = 0x5000001, Remote IP addr = 0x0,
       Local RTP port = 16406, Remote RTP port = 0, mode = 0x2
*Mar  1 01:01:51.598:rtpspi_start_rtcp_session:RTP Session creation Success.
*Mar  1 01:01:51.598:rtpspi_call_setup:RTP Session creation Success.
*Mar  1 01:01:51.598:rtpspi_call_setup:calling cc_api_call_connected()
*Mar  1 01:01:51.598:rtpspi_modify_rtcp_session_parameters():xmit fn=0xDA7494,
dstIF=0x1964EEC, dstCallID=10, voip_mode=0x2, rtp_mode=0x2, ssrc_status=0
*Mar  1 01:01:51.598:rtpspi_bridge:Calling cc_api_bridge_done() for 11(0x1AF5400) and
10(0x0).
*Mar  1 01:01:51.602:rtpspi_caps_ind:caps from VTSP:codec=0x83FB, codec_bytes=0x50,
                  fax rate=0x7F, vad=0x3 modem=0x0
*Mar  1 01:01:51.602:rtpspi_get_rtcp_session_parameters():CURRENT VALUES:
dstIF=0x1964EEC, dstCallID=10, current_seq_num=0x0
*Mar  1 01:01:51.602:rtpspi_get_rtcp_session_parameters():NEW VALUES:
dstIF=0x1964EEC, dstCallID=10, current_seq_num=0xF1E
*Mar  1 01:01:51.602:rtpspi_caps_ind:Caps Used:codec=0x1, codec bytes=80,
       fax rate=0x1, vad=0x1, modem=0x1, dtmf_relay=0x1, seq_num_start=0xF1F
*Mar  1 01:01:51.602:rtpspi_caps_ind:calling cc_api_caps_ind().
*Mar  1 01:01:51.822:rtpspi_do_call_modify:Remote RTP port changed. New port=16498
*Mar  1 01:01:51.822:rtpspi_do_call_modify:Remote IP addrs changed. New IP addrs=0x6000001
*Mar  1 01:01:51.822:rtpspi_do_call_modify:Starting new RTCP session.
*Mar  1 01:01:51.822:rtpspi_start_rtcp_session:Removing old RTCP session.
```

```
*Mar  1 01:01:51.822:rtpspi_start_rtcp_session:Starting RTCP session.
        Local IP addr = 0x5000001, Remote IP addr = 0x6000001,
        Local RTP port = 16406, Remote RTP port = 16498, mode = 0x2
*Mar  1 01:01:51.822:rtpspi_start_rtcp_session:RTCP Timer creation Success. (5)*(5000)
*Mar  1 01:01:51.826:rtpspi_start_rtcp_session:RTP Session creation Success.
*Mar  1 01:01:51.826:rtpspi_do_call_modify:RTP Session creation Success.
*Mar  1 01:01:51.826:rtpspi_do_call_modify:Calling cc_api_call_modify(), result=0x0
*Mar  1 01:01:57.296:rtpspi_do_call_modify:Mode changed. new = 3, old = 2
*Mar  1 01:01:57.296:rtpspi_modify_rtcp_session_parameters():xmit fn=0xDA7494,
dstIF=0x1964EEC, dstCallID=10, voip_mode=0x3, rtp_mode=0x3, ssrc_status=2
*Mar  1 01:01:57.296:rtpspi_do_call_modify:RTCP Timer start.
*Mar  1 01:01:57.296:rtpspi_do_call_modify:Calling cc_api_call_modify(), result=0x0
*Mar  1 01:03:06.108:rtpspi_modify_rtcp_session_parameters():xmit fn=0x0,
dstIF=0x0, dstCallID=0, voip_mode=0x3, rtp_mode=0x3, ssrc_status=2
*Mar  1 01:03:06.112:rtpspi_do_call_disconnect:calling rtpspi_call_cleanup(). call-id=11
*Mar  1 01:03:06.112:rtpspi_call_cleanup:releasing ccb cache. RTP port=16406
*Mar  1 01:03:06.112:rtpspi_call_cleanup:RTCP Timer Stop.
*Mar  1 01:03:06.112:rtpspi_call_cleanup:deallocating RTP port 16406.
*Mar  1 01:03:06.112::rtpspi_call_cleanup freeing ccb (0x1AF5400)
```

Related Commands

Command	Description
debug rtpspi all	Debugs all RTP SPI errors, sessions, and in/out functions.
debug rtpspi errors	Debugs RTP SPI errors.
debug rtpspi inout	Debugs RTP SPI in/out functions.
debug rtpspi send-nse	Triggers the RTP SPI to send a triple redundant NSE.
debug sgcp errors	Debugs SGCP errors.
debug sgcp events	Debugs SGCP events.
debug sgcp packet	Debugs SGCP packets.
sgcp	Starts and allocates resources for the SCGP daemon.
debug vtsp send-nse	Sends and debugs a triple redundant NSE from the DSP to a remote gateway.

debug rtr error

Note Effective with Cisco IOS Release 12.2(31)SB2, the **debugrtrerror**command is replaced by the **debugipslamonitorerror**command. Effective with Cisco IOS Release 12.2(33)SRB, the **debugrtrerror**command is replaced by the **debugipslaerror**command. See the **debugipslamonitorerror** and **debugipslaerror** commands for more information.

To enable debugging output of Cisco IOS IP Service Level Agreements (SLAs) operation run-time errors, use the **debugrtrerror**command in privileged EXEC mode. To disable debugging output, use the **no**form of this command.

debug rtr error [*operation-number*]
no debug rtr error [*operation-number*]

Syntax Description

operation-number	(Optional) Identification number of the operation for which debugging output is to be enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.0(5)T	This command was modified.
12.3(14)T	This command was replaced by the **debugipslamonitorerror**command.
12.2(31)SB2	This command was replaced by the **debugipslamonitorerror**command.
12.2(33)SRB	This command was replaced by the **debugipslaerror**command.

Usage Guidelines

The **debugrtrerror**command displays run-time errors. When an operation number other than 0 is specified, all run-time errors for that operation are displayed when the operation is active. When the operation number is 0, all run-time errors relating to the IP SLAs scheduler process are displayed. When no operation number is specified, all run-time errors for all active operations configured on the router are displayed.

Note Use the **debugrtrerror**command before using the **debugrtrtrace** command because the **debugrtrerror**command generates a lesser amount of debugging output.

Examples

The following is sample output from the **debugrtrerror** command. The output indicates failure because the target is not there or because the responder is not enabled on the target. All debugging output for IP SLAs (including the output from the **debugrtrtrace** command) has the following format.

```
Router# debug rtr error
```

```
May  5 05:00:35.483: control message failure:1
May  5 05:01:35.003: control message failure:1
May  5 05:02:34.527: control message failure:1
May  5 05:03:34.039: control message failure:1
May  5 05:04:33.563: control message failure:1
May  5 05:05:33.099: control message failure:1
May  5 05:06:32.596: control message failure:1
May  5 05:07:32.119: control message failure:1
May  5 05:08:31.643: control message failure:1
May  5 05:09:31.167: control message failure:1
May  5 05:10:30.683: control message failure:1
```

The following table describes the significant fields shown in the display.

Table 87: debug rtr error Field Descriptions

Field	Description
RTR 1	Number of the operation generating the message.
Error Return Code	Message identifier indicating the error type (or error itself).
LU0 RTR Probe 1	Name of the process generating the message.
in echoTarget on call luReceive LuApiReturnCode of InvalidHandle - invalid host name or API handle	Supplemental messages that pertain to the message identifier.

Related Commands

Command	Description
debug rtr trace	Traces the execution of an IP SLAs operation.

debug rtr mpls-lsp-monitor

Note Effective with Cisco IOS Release 12.2(31)SB2, the **debug rtr mpls-lsp-monitor**command is replaced by the **debug ip sla monitor mpls-lsp-monitor**command. Effective with Cisco IOS Release 12.2(33)SRB, the **debug rtr mpls-lsp-monitor**command is replaced by the **debug ip sla mpls-lsp-monitor**command. See the **debug ip sla monitor mpls-lsp-monitor** and **debug ip sla mpls-lsp-monitor** commands for more information.

To enable debugging output for the IP Service Level Agreements (SLAs) label switched path (LSP) Health Monitor, use the **debug rtr mpls-lsp-monitor**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtr mpls-lsp-monitor [*operation-number*]
no debug rtr mpls-lsp-monitor [*operation-number*]

Syntax Description

operation-number	(Optional) Number of the LSP Health Monitor operation for which the debugging output will be displayed.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(27)SBC	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(31)SB2	This command was replaced by the **debug ip sla monitor mpls-lsp-monitor**command.
12.2(33)SRB	This command was replaced by the **debug ip sla mpls-lsp-monitor**command.

Examples

The following is sample output from the **debug rtr mpls-lsp-monitor** command. This output shows that three VPNs associated with router 10.10.10.8 (red, blue, and green) were discovered and that this information was added to the LSP Health Monitor scan queue. Also, since router 10.10.10.8 is a newly discovered Border Gateway Protocol (BGP) next hop neighbor, a new IP SLAs operation for router 10.10.10.8 (Probe 100005) is being created and added to the LSP Health Monitor multioperation schedule. Even though router 10.10.10.8 belongs to three VPNs, only one IP SLAs operation is being created.

```
Router# debug rtr mpls-lsp-monitor
SAA MPLSLM debugging for all entries is on
*Aug 19 19:59: SAA MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: SAA MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: SAA MPLSLM(1):Next hop 10.10.10.8 added in AddQ
*Aug 19 19:59: SAA MPLSLM(1):Adding vrf red into tree entry 10.10.10.8
*Aug 19 19:59: SAA MPLSLM(1):Adding Probe 100005
*Aug 19 19:59: SAA MPLSLM(1):Adding ProbeID 100005 to tree entry 10.10.10.8 (1)
```

```
*Aug 19 19:59: SAA MPLSLM(1):Adding vrf blue into tree entry 10.10.10.8
*Aug 19 19:59: SAA MPLSLM(1):Duplicate in AddQ 10.10.10.8
*Aug 19 19:59: SAA MPLSLM(1):Adding vrf green into tree entry 10.10.10.8
*Aug 19 19:59: SAA MPLSLM(1):Duplicate in AddQ 10.10.10.8
*Aug 19 19:59: SAA MPLSLM(1):Added Probe(s) 100005 will be scheduled after 26 secs over
schedule period 60
```

Related Commands

Command	Description
rtr mpls-lsp-monitor	Begins configuration for an IP SLAs LSP Health Monitor operation and enters SAA MPLS configuration mode.

debug rtr trace

Note Effective with Cisco IOS Release 12.2(31)SB2, the **debug rtr trace**command is replaced by the **debug ip sla monitor trace**command. Effective with Cisco IOS Release 12.2(33)SRB, the **debug rtr trace**command is replaced by the **debug ip sla trace**command. See the **debug ip sla monitor trace** and **debug ip sla trace** commands for more information.

To trace the execution of a Cisco IOS IP Service Level Agreements (SLAs) operation, use the **debug rtr trace**command in privileged EXEC mode. To disable trace debugging output, use the **no**form of this command.

debug rtr trace [*operation-number*]
no debug rtr trace [*operation-number*]

Syntax Description

operation-number	(Optional) Identification number of the operation for which debugging output is to be enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.0(5)T	This command was modified.
12.3(14)T	This command was replaced by the **debug ip sla monitor trace**command.
12.2(31)SB2	This command was replaced by the **debug ip sla monitor trace**command.
12.2(33)SRB	This command was replaced by the **debug ip sla trace**command.

Usage Guidelines

When an operation number other than 0 is specified, execution for that operation is traced. When the operation number is 0, the IP SLAs scheduler process is traced. When no operation number is specified, all active operations are traced.

The **debug rtr trace** command also enables **debug rtr error** command for the specified operation. However, the **no debug rtr trace** command does not disable the **debug rtr error** command. You must manually disable the command by using the **no debug rtr error** command.

All debuggng output (including **debug rtr error** command output) has the format shown in the **debug rtr error** command output example.

Note The **debug rtr trace**command can generate a large number of debug messages. First use the **debug rtr error** command, and then use the **debug rtr trace** on a per-operation basis.

Examples

The following is sample output from the **debug rtr trace** command. In this example, an operation is traced through a single operation attempt: the setup of a connection to the target, and the attempt at an echo to calculate UDP packet response time.

```
Router# debug rtr trace
Router# RTR 1:Starting An Echo Operation - IP RTR Probe 1
May  5 05:25:08.584:rtt hash insert :3.0.0.3 3383
May  5 05:25:08.584:    source=3.0.0.3(3383)  dest-ip=5.0.0.1(9)
May  5 05:25:08.588:sending control msg:
May  5 05:25:08.588: Ver:1 ID:51 Len:52
May  5 05:25:08.592:cmd:command:RTT_CMD_UDP_PORT_ENABLE, ip:5.0.0.1, port:9, duration:5000
May  5 05:25:08.607:receiving reply
May  5 05:25:08.607: Ver:1 ID:51 Len:8
May  5 05:25:08.623:    local delta:8
May  5 05:25:08.627:    delta from responder:1
May  5 05:25:08.627:    received <16> bytes and      responseTime = 3 (ms)
May  5 05:25:08.631:rtt hash remove:3.0.0.3 3383RTR 1:Starting An Echo Operation - IP RTR
Probe 1
May  5 05:26:08.104:rtt hash insert :3.0.0.3 2974
May  5 05:26:08.104:    source=3.0.0.3(2974)  dest-ip=5.0.0.1(9)
May  5 05:26:08.108:sending control msg:
May  5 05:26:08.108: Ver:1 ID:52 Len:52
May  5 05:26:08.112:cmd:command:RTT_CMD_UDP_PORT_ENABLE, ip:5.0.0.1, port:9, duration:5000
May  5 05:26:08.127:receiving reply
May  5 05:26:08.127: Ver:1 ID:52 Len:8
May  5 05:26:08.143:    local delta:8
May  5 05:26:08.147:    delta from responder:1
May  5 05:26:08.147:    received <16> bytes and      responseTime = 3 (ms)
May  5 05:26:08.151:rtt hash remove:3.0.0.3 2974RTR 1:Starting An Echo Operation - IP RTR
Probe 1
```

Related Commands

Command	Description
debug rtr error	Enables debugging output of IP SLAs operation run-time errors.

debug rtsp

To show the status of the Real-Time Streaming Protocol (RTSP) client or server, use the **debug rtsp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp *type* [{**all** | **api** | **error** | **pmh** | **session** | **socket**}]
[**no**] **debug rtsp** *type* [{**all** | **api** | **pmh** | **session** | **socket**}]

Syntax Description

type	Type of debug messages to display. The keywords are as follows: • all--(Optional) Displays debug output for all clients or servers. • api--(Optional) Displays debug output for the client or server API. • error--(Optional) Displays errors when they are errors otherwise no output is displayed. • pmh--(Optional) Displays debug output for the Protocol Message Handler (PMH). • session--(Optional) Displays debug output for the client or server session. • socket--(Optional) Displays debug output for the client or server socket data.

Command Default

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	The new debug header was added to the following Cisco routers: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660 series; on the following universal gateways: Cisco AS5350, Cisco AS5400, and Cisco AS5850; on the following access servers: Cisco AS5300, and Cisco AS5800; and, on the Cisco MC3810 multiservice access concentrators.

Examples

The following is sample output that displays when the **debug rtsp command is entered with the api**keyword:

```
Router# debug rtsp api
!
RTSP client API debugging is on
!
Jan  1 00:23:15.775:rtsp_api_create_session:sess_id=0x61A07C78,evh=0x60D6E62C
context=0x61A07B28
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C2B10C
Jan  1 00:23:15.775:rtsp_api_handle_req_set_params:msg=0x61C2B10C
Jan  1 00:23:15.775:rtsp_api_free_msg_buffer:msg=0x61C2B10C
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C293CC
Jan  1 00:23:15.775:rtsp_api_handle_req_set_params:msg=0x61C293CC
Jan  1 00:23:15.775:rtsp_api_free_msg_buffer:msg=0x61C293CC
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C2970C
```

```
Jan  1 00:23:15.775:rtsp_api_handle_req_set_params:msg=0x61C2970C
Jan  1 00:23:15.775:rtsp_api_free_msg_buffer:msg=0x61C2970C
!
Jan  1 00:23:15.775:rtsp_api_request:msg=0x61C29A4C
!
Jan  1 00:23:22.099:rtsp_api_free_msg_buffer:msg=0x61C29A4C
Jan  1 00:23:22.115:rtsp_api_request:msg=0x61C2A40C
Jan  1 00:23:22.115:rtsp_api_free_msg_buffer:msg=0x61C2A40C
```

Related Commands

Command	Description
debug rtsp api	Displays debug output for the RTSP client API.
debug rtsp client session	Displays debug output for the RTSP client data.
debug rtsp socket	Displays debug output for the RTSP client socket data.

debug rtsp all

To display all related information about the Real Time Streaming Protocol (RTSP) data, use the **debugrtspall**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp all
no debug rtsp all

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.

Usage Guidelines

We recommend that you log output from the **debugrtspall** command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Examples

The following example shows debugging output for the**debugrtspall**command. The **showdebug** command shows which RTSP modules are traced.

```
Router# debug rtsp all
All RTSP client debugging is on
Router# show debug
RTSP:
  RTSP client Protocol Error debugging is on
  RTSP client Protocol Message Handler debugging is on
  RTSP client API debugging is on
  RTSP client socket debugging is on
  RTSP client session debugging is on
Router#
Router#!call initiated
Router#
*Mar 11 03:14:23.471: //-1//RTSP:/rtsp_get_new_scb:
*Mar 11 03:14:23.471: //-1//RTSP:/rtsp_initialize_scb:
*Mar 11 03:14:23.471: //-1//RTSP:/rtsplib_init_svr_session: 0x63A5FE6C
*Mar 11 03:14:23.471: //-1//RTSP:/rtsp_api_create_session: evh=0x6155F0D4 context=0x6345042C
*Mar 11 03:14:23.471: //-1//RTSP:/rtsp_get_new_scb:
*Mar 11 03:14:23.471: //-1//RTSP:/rtsp_initialize_scb:
*Mar 11 03:14:23.471: //-1//RTSP:/rtsplib_init_svr_session: 0x63A5D874
*Mar 11 03:14:23.471: //-1//RTSP:/rtsp_api_create_session: evh=0x6155F204 context=0x6345046C
*Mar 11 03:14:23.471: //-1//RTSP:RS45:/rtsp_api_request: msg=0x63A59FB8
*Mar 11 03:14:23.471: //-1//RTSP:RS45:/rtsp_api_handle_req_set_params: msg=0x63A59FB8
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_free_msg_buffer: msg=0x63A59FB8
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_request: msg=0x63A5A304
```

```
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_handle_req_set_params: msg=0x63A5A304
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_free_msg_buffer: msg=0x63A5A304
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_request: msg=0x63A5A650
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_handle_req_set_params: msg=0x63A5A650
*Mar 11 03:14:23.475: //166//RTSP:LP:RS45:/rtsp_api_handle_req_set_params:
*Mar 11 03:14:23.475: //-1//RTSP:RS45:/rtsp_api_free_msg_buffer: msg=0x63A5A650
*Mar 11 03:14:23.475: //-1//RTSP:RS46:/rtsp_api_request: msg=0x63A5A99C
*Mar 11 03:14:23.475: //-1//RTSP:RS46:/rtsp_api_handle_req_set_params: msg=0x63A5A99C
*Mar 11 03:14:23.475: //166//RTSP:LP:RS46:/rtsp_api_handle_req_set_params:
*Mar 11 03:14:23.475: //-1//RTSP:RS46:/rtsp_api_free_msg_buffer: msg=0x63A5A99C
Router#
Router#!call answered
Router#
Router#!digits dialed
Router#
Router#!call terminated
Router#
*Mar 11 03:14:51.603: //-1//RTSP:RS45:/rtsp_api_request: msg=0x63A5ACE8
*Mar 11 03:14:51.603: //-1//RTSP:RS46:/rtsp_api_request: msg=0x63A5B034
*Mar 11 03:14:51.607: //-1//RTSP:RS45:/rtsp_control_process_msg:
*Mar 11 03:14:51.607: //166//RTSP:/rtsp_control_process_msg: received MSG request of TYPE
0
*Mar 11 03:14:51.607: //166//RTSP:/rtsp_set_event: api_req_msg_type=RTSP_API_REQ_DESTROY
*Mar 11 03:14:51.607: //166//RTSP:/rtsp_session_cleanup:
*Mar 11 03:14:51.607: //-1//RTSP:/rtsplib_free_svr_session:
*Mar 11 03:14:51.607: //-1//RTSP:/rtsplib_stop_timer: timer(0x638D5DDC) stops
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_create_session_history: scb=0x63A5FE6C, callID=0xA6
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_create_session_history: No streams in session control
 block
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_session_cleanup: deleting session: scb=0x63A5FE6C
*Mar 11 03:14:51.611: //-1//RTSP:RS45:/rtsp_api_free_msg_buffer: msg=0x63A5ACE8
*Mar 11 03:14:51.611: //-1//RTSP:RS46:/rtsp_control_process_msg:
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_control_process_msg: received MSG request of TYPE
0
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_set_event: api_req_msg_type=RTSP_API_REQ_DESTROY
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_session_cleanup:
*Mar 11 03:14:51.611: //-1//RTSP:/rtsplib_free_svr_session:
*Mar 11 03:14:51.611: //-1//RTSP:/rtsplib_stop_timer: timer(0x63A60110) stops
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_create_session_history: scb=0x63A5D874, callID=0xA6
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_create_session_history: No streams in session control
 block
*Mar 11 03:14:51.611: //166//RTSP:/rtsp_session_cleanup: deleting session: scb=0x63A5D874
*Mar 11 03:14:51.611: //-1//RTSP:RS46:/rtsp_api_free_msg_buffer: msg=0x63A5B034
```

The following table describes the significant fields shown in the display.

Table 88: debug rtsp all Field Descriptions

Field	Description
//-1/	Indicates that the CallEntry ID for the module is unavailable.
//166/	Identifies the CallEntry ID.
RTSP:	Identifies the RTSP module.
rtsp_*functionname*	Identifies the function name.

Related Commands

Command	Description
debug rtsp api	Displays debugging output for the RTSP client API.
debug rtsp error	Displays error message for RTSP data.
debug rtsp pmh	Displays debugging messages for the PMH.
debug rtsp socket	Displays debugging output for the RTSP client socket data.
voice call debug	Allows configuration of the voice call debugging output.

debug rtsp api

To display information about the Real Time Streaming Protocol (RTSP) application programming interface (API) messages passed down to the RTSP client, use the **debugrtspapi**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp api
no debug rtsp api

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.

Usage Guidelines

We recommend that you log output from the **debugrtspapi** command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Examples

The following example shows output from the**debugrtspapi**command:

```
Router# debug rtsp api
RTSP client API debugging is on
Router# !call initiated
*Mar 11 03:04:41.699: //-1//RTSP:/rtsp_api_create_session: evh=0x6155F0D4 context=0x6345088C
*Mar 11 03:04:41.699: //-1//RTSP:/rtsp_api_create_session: evh=0x6155F204 context=0x634508CC
*Mar 11 03:04:41.699: //-1//RTSP:RS35:/rtsp_api_request: msg=0x63A59FB8
*Mar 11 03:04:41.699: //-1//RTSP:RS35:/rtsp_api_handle_req_set_params: msg=0x63A59FB8
*Mar 11 03:04:41.699: //-1//RTSP:RS35:/rtsp_api_free_msg_buffer: msg=0x63A59FB8
*Mar 11 03:04:41.699: //-1//RTSP:RS35:/rtsp_api_request: msg=0x63A5A304
*Mar 11 03:04:41.699: //-1//RTSP:RS35:/rtsp_api_handle_req_set_params: msg=0x63A5A304
*Mar 11 03:04:41.699: //-1//RTSP:RS35:/rtsp_api_free_msg_buffer: msg=0x63A5A304
*Mar 11 03:04:41.703: //-1//RTSP:RS35:/rtsp_api_request: msg=0x63A5A650
*Mar 11 03:04:41.703: //-1//RTSP:RS35:/rtsp_api_handle_req_set_params: msg=0x63A5A650
*Mar 11 03:04:41.703: //146//RTSP:LP:RS35:/rtsp_api_handle_req_set_params:
*Mar 11 03:04:41.703: //-1//RTSP:RS35:/rtsp_api_free_msg_buffer: msg=0x63A5A650
*Mar 11 03:04:41.703: //-1//RTSP:RS36:/rtsp_api_request: msg=0x63A5A99C
*Mar 11 03:04:41.703: //-1//RTSP:RS36:/rtsp_api_handle_req_set_params: msg=0x63A5A99C
*Mar 11 03:04:41.703: //146//RTSP:LP:RS36:/rtsp_api_handle_req_set_params:
*Mar 11 03:04:41.703: //-1//RTSP:RS36:/rtsp_api_free_msg_buffer: msg=0x63A5A99C
Router!call answered
Router#!digits dialed
Router#!call terminated
*Mar 11 03:05:15.367: //-1//RTSP:RS35:/rtsp_api_request: msg=0x63A5ACE8
*Mar 11 03:05:15.367: //-1//RTSP:RS36:/rtsp_api_request: msg=0x63A5B034
*Mar 11 03:05:15.367: //-1//RTSP:RS35:/rtsp_api_free_msg_buffer: msg=0x63A5ACE8
*Mar 11 03:05:15.367: //-1//RTSP:RS36:/rtsp_api_free_msg_buffer: msg=0x63A5B034
```

The following table describes the significant fields shown in the display.

Table 89: debug rtsp api Field Descriptions

Field	Description
//-1/	Indicates that the CallEntry ID for the module is unavailable.
//146/	Identifies the CallEntry ID.
RTSP:	Identifies the RTSP module.
rtsp_*functionname*	Identifies the function name.

Related Commands

Command	Description
debug rtsp error	Displays error message for RTSP data.
debug rtsp pmh	Displays debugging messages for the PMH.
debug rtsp socket	Displays debugging output for the RTSP client socket data.
voice call debug	Allows configuration of the voice call debugging output.

debug rtsp client

Note Effective with Release 12.3(4), the **debug rtsp cleint** command is replaced by the **debug rtsp session** command. See the **debug rtsp session** command for more information.

To display client information and stream information for the stream that is currently active for the Real Time Streaming Protocol (RTSP) client, use the **debug rtsp client**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp client
no debug rtsp client

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.3(4)T	This command was replaced by the **debug rtsp session** command.

Usage Guidelines

We recommend that you log output from the **debug rtsp client** command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Related Commands

Command	Description
debug rtsp api	Displays debugging output for the RTSP client API.
debug rtsp error	Displays error message for RTSP data.
debug rtsp pmh	Displays debugging messages for the PMH.
debug rtsp socket	Displays debugging output for the RTSP client socket data.
voice call debug	Allows configuration of the voice call debugging output.

debug rtsp client session

Note Effective with Release 12.3(4), the **debug rtsp cleint session**command is replaced by the **debug rtsp session** command. See the **debug rtsp session** command for more information.

To display debug messages about the Real Time Streaming Protocol (RTSP) client or the current session, use the **debug rtsp**command. To disable debugging output, use the **no** form of this command.

debug rtsp [{**client** | **session**}]
no debug rtsp [{**client** | **session**}]

Syntax Description

client	(Optional) Displays client information and stream information for the stream that is currently active.
session	(Optional) Displays cumulative information about the session, packet statistics, and general call information such as call ID, session ID, individual RTSP stream URLs, packet statistics, and play duration.

Command Default

Debug is not enabled.

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.3(4)T	This command was replaced by the **debug rtsp session** command.

Examples

The following example displays the debug messages of the RTSP session:

```
Router# debug rtsp session
RTSP client session debugging is on
router#
Jan  1 00:08:36.099:rtsp_get_new_scb:
Jan  1 00:08:36.099:rtsp_initialize_scb:
Jan  1 00:08:36.099:rtsp_control_process_msg:
Jan  1 00:08:36.099:rtsp_control_process_msg:received MSG request of TYPE 0
Jan  1 00:08:36.099:rtsp_set_event:
Jan  1 00:08:36.099:rtsp_set_event:api_req_msg_type=RTSP_API_REQ_PLAY
Jan  1 00:08:36.103:rtsp_set_event:url:[rtsp://rtsp-cisco.cisco.com:554/en_welcome.au]
Jan  1 00:08:36.103:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:36.103:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_IDLE
                    rtsp_event = RTSP_EV_PLAY_OR_REC
Jan  1 00:08:36.103:act_idle_event_play_or_rec_req:
Jan  1 00:08:36.103:rtsp_resolve_dns:
Jan  1 00:08:36.103:rtsp_resolve_dns:IP Addr = 1.13.79.6:
Jan  1 00:08:36.103:rtsp_connect_to_svr:
Jan  1 00:08:36.103:rtsp_connect_to_svr:socket=0, connection_state = 2
Jan  1 00:08:36.103:rtsp_start_timer:timer (0x62128FD0)starts - delay (10000)
Jan  1 00:08:36.107:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:36.107:rtsp_stop_timer:timer(0x62128FD0) stops
Jan  1 00:08:36.107:rtsp_process_async_event:SCB=0x62128F08
```

```
Jan  1 00:08:36.107:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_IDLE
                    rtsp_event = RTSP_EV_SVR_CONNECTED
Jan  1 00:08:36.107:act_idle_event_svr_connected:
Jan  1 00:08:36.107:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:36.783:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:36.783:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:36.783:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_SVR_DESC_OR_ANNOUNCE_RESP
Jan  1 00:08:36.783:act_ready_event_desc_or_announce_resp:
Jan  1 00:08:36.783:act_ready_event_desc_or_announce_resp:RTSP_STATUS_DESC_OR_ANNOUNCE_RESP_OK
Jan  1 00:08:37.287:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:37.287:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:37.287:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_SVR_SETUP_RESP
Jan  1 00:08:37.287:act_ready_event_setup_resp:
Jan  1 00:08:37.287:act_ready_event_setup_resp:Remote RTP Port=13344
Jan  1 00:08:37.287:rtsp_rtp_stream_setup:scb=0x62128F08, callID=0x7 record=0
Jan  1 00:08:37.287:rtsp_rtp_stream_setup:Starting RTCP session.
       Local IP addr = 1.13.79.45, Remote IP addr = 1.13.79.6,
       Local RTP port = 18748, Remote RTP port = 13344 CallID=8
Jan  1 00:08:37.291:xmit_func = 0x0 vdbptr = 0x61A0FC98
Jan  1 00:08:37.291:rtsp_control_main:CCAPI Queue Event
Jan  1 00:08:37.291:rtsp_rtp_associate_done:ev=0x62070E08, callID=0x7
Jan  1 00:08:37.291:rtsp_rtp_associate_done:scb=0x62128F08
Jan  1 00:08:37.291:rtsp_rtp_associate_done:callID=0x7, pVdb=0x61F4FBC8,
Jan  1 00:08:37.291:                  spi_context=0x6214145C
Jan  1 00:08:37.291:                  disposition=0, playFunc=0x60CA2238,
Jan  1 00:08:37.291:                  codec=0x5, vad=0, mediaType=6,
Jan  1 00:08:37.291:                  stream_assoc_id=1
Jan  1 00:08:37.291:rtsp_rtp_modify_session:scb=0x62128F08, callID=0x7
Jan  1 00:08:37.291:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:37.291:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_ASSOCIATE_DONE
Jan  1 00:08:37.291:act_ready_event_associate_done:
Jan  1 00:08:37.291:rtsp_get_stream:
Jan  1 00:08:37.783:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:37.783:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:37.783:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_READY
                    rtsp_event = RTSP_EV_SVR_PLAY_OR_REC_RESP
Jan  1 00:08:37.783:act_ready_event_play_or_rec_resp:
Jan  1 00:08:37.783:rtsp_start_timer:timer (0x62128FB0)starts - delay (4249)
rtsp-5#
Jan  1 00:08:42.035:rtsp_process_timer_events:
Jan  1 00:08:42.035:rtsp_process_timer_events:PLAY OR RECORD completed
Jan  1 00:08:42.035:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:42.035:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_PLAY_OR_REC
                    rtsp_event = RTSP_EV_PLAY_OR_REC_TIMER_EXPIRED
Jan  1 00:08:42.035:act_play_event_play_done:
Jan  1 00:08:42.035:act_play_event_play_done:elapsed play time = 4249 total play time =
4249
Jan  1 00:08:42.035:rtsp_send_teardown_to_svr:
Jan  1 00:08:42.487:rtsp_control_main:SOCK= 0 Event=0x1
Jan  1 00:08:42.487:rtsp_process_async_event:SCB=0x62128F08
Jan  1 00:08:42.487:rtsp_process_async_event:rtsp_state = RTSP_SES_STATE_PLAY_OR_REC
                    rtsp_event = RTSP_EV_SVR_TEARDOWN_RESP
Jan  1 00:08:42.487:act_play_event_teardown_resp:
Jan  1 00:08:42.487:rtsp_server_closed:
Jan  1 00:08:42.487:rtsp_send_resp_to_api:
Jan  1 00:08:42.487:rtsp_send_resp_to_api:sending RESP=RTSP_STATUS_PLAY_COMPLETE
Jan  1 00:08:42.491:rtsp_rtp_teardown_stream:scb=0x62128F08, callID=0x7
Jan  1 00:08:42.491:rtsp_rtp_stream_cleanup:scb=0x62128F08, callID=0x7
Jan  1 00:08:42.491:rtsp_update_stream_stats:scb=0x62128F08, stream=0x61A43350,
Jan  1 00:08:42.491:call_info=0x6214C67C, callID=0x7
Jan  1 00:08:42.491:rtsp_update_stream_stats:rx_bytes = 25992
```

```
Jan  1 00:08:42.491:rtsp_update_stream_stats:rx_packetes = 82
Jan  1 00:08:42.491:rtsp_reinitialize_scb:
Jan  1 00:08:42.503:rtsp_control_process_msg:
Jan  1 00:08:42.503:rtsp_control_process_msg:received MSG request of TYPE 0
Jan  1 00:08:42.503:rtsp_set_event:
Jan  1 00:08:42.503:rtsp_set_event:api_req_msg_type=RTSP_API_REQ_DESTROY
Jan  1 00:08:42.503:rtsp_session_cleanup:
Jan  1 00:08:42.503:rtsp_create_session_history:scb=0x62128F08, callID=0x7
Jan  1 00:08:42.503:rtsp_insert_session_history_record:current=0x6214BDC8, callID=0x7
Jan  1 00:08:42.503:rtsp_insert_session_history_record:count = 3
Jan  1 00:08:42.503:rtsp_insert_session_history_record:starting history record deletion_timer
 of10 minutes
Jan  1 00:08:42.503:rtsp_session_cleanup:deleting session:scb=0x62128F08
Router#
```

Related Commands

Command	Description
debug rtsp all	Displays debugging output for the RTSP client API.
debug rtsp pmh	Displays debugging messages for the PMH.
debug rtsp socket	Displays debugging output for the RTSP client socket data.

debug rtsp error

To display error information about the Real-Time Streaming Protocol (RTSP) client, use the **debug rtsp error**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp error
no debug rtsp error

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.

Usage Guidelines

We recommend that you log output from the **debug rtsp error**command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Related Commands

Command	Description
debug rtsp api	Displays debugging output for the RTSP client API.
debug rtsp pmh	Displays debugging messages for the PMH.
debug rtsp socket	Displays debugging output for the RTSP client socket data.
voice call debug	Allows configuration of the voice call debugging output.

debug rtsp pmh

To display debugging information about the Protocol Message Handler (PMH), use the **debug rtsp pmh**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp pmh
no debug rtsp pmh

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.

Usage Guidelines

We recommend that you log output from the **debug rtsp pmh** command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Related Commands

Command	Description
debug rtsp api	Displays debugging output for the RTSP client API.
debug rtsp error	Displays error message for RTSP data.
debug rtsp socket	Displays debugging output for the RTSP client socket data.
voice call debug	Allows configuration of the voice call debugging output.

debug rtsp session

To display client information and stream information for the stream that is currently active for the Real Time Streaming Protocol (RTSP) client, use the **debugrtspsession**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp session
no debug rtsp session

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.
12.3(4)T	This command replaces the **debugrtspclient** command and the **debugrtspclientsession** command.

Usage Guidelines

We recommend that you log output from the **debugrtspsession** command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Examples

The following example shows the display of the debugging messages of the RTSP session:

```
Router# debug rtsp session
RTSP client session debugging is on
Router#
Router#!call initiated
Router#
*Mar 11 03:09:58.123: //-1//RTSP:/rtsp_get_new_scb:
*Mar 11 03:09:58.123: //-1//RTSP:/rtsp_initialize_scb:
*Mar 11 03:09:58.123: //-1//RTSP:/rtsplib_init_svr_session: 0x63A5FE6C
*Mar 11 03:09:58.123: //-1//RTSP:/rtsp_get_new_scb:
*Mar 11 03:09:58.123: //-1//RTSP:/rtsp_initialize_scb:
*Mar 11 03:09:58.123: //-1//RTSP:/rtsplib_init_svr_session: 0x63A5D874
Router#
Router#!call answered
Router#
Router#!digits dialed
Router#
Router#!call terminated
Router#
*Mar 11 03:10:38.139: //-1//RTSP:RS41:/rtsp_control_process_msg:
*Mar 11 03:10:38.139: //158//RTSP:/rtsp_control_process_msg: received MSG request of TYPE
0
*Mar 11 03:10:38.139: //158//RTSP:/rtsp_set_event: api_req_msg_type=RTSP_API_REQ_DESTROY
*Mar 11 03:10:38.139: //158//RTSP:/rtsp_session_cleanup:
```

```
*Mar 11 03:10:38.139: //-1//RTSP:/rtsplib_free_svr_session:
*Mar 11 03:10:38.139: //-1//RTSP:/rtsplib_stop_timer: timer(0x638D5DDC) stops
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_create_session_history: scb=0x63A5FE6C, callID=0x9E
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_create_session_history: No streams in session control
 block
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_session_cleanup: deleting session: scb=0x63A5FE6C
*Mar 11 03:10:38.143: //-1//RTSP:RS42:/rtsp_control_process_msg:
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_control_process_msg: received MSG request of TYPE
0
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_set_event: api_req_msg_type=RTSP_API_REQ_DESTROY
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_session_cleanup:
*Mar 11 03:10:38.143: //-1//RTSP:/rtsplib_free_svr_session:
*Mar 11 03:10:38.143: //-1//RTSP:/rtsplib_stop_timer: timer(0x63A60110) stops
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_create_session_history: scb=0x63A5D874, callID=0x9E
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_create_session_history: No streams in session control
 block
*Mar 11 03:10:38.143: //158//RTSP:/rtsp_session_cleanup: deleting session: scb=0x63A5D874
```

The following table describes the significant fields shown in the display.

Table 90: debug rtsp session Field Descriptions

Field	Description
//-1/	Indicates that the CallEntry ID for the module is unavailable.
//158/	Identifies the CallEntry ID.
RTSP:	Identifies the RTSP module.
rtsp_*functionname*	Identifies the function name.

Related Commands

Command	Description
debug rtsp api	Displays debugging output for the RTSP client API.
debug rtsp error	Displays error message for RTSP data.
debug rtsp pmh	Displays debugging messages for the PMH.
debug rtsp socket	Displays debugging output for the RTSP client socket data.
voice call debug	Allows configuration of the voice call debugging output.

debug rtsp socket

To display debugging messages about the packets received or sent on the TCP or User Datagram Protocol (UDP) sockets, use the **debug rtsp socket**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug rtsp socket
no debug rtsp socket

Syntax Description

This command has no arguments or keywords.

Command Default

Debug is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(11)T	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3620, Cisco 3640, and Cisco 3660, Cisco AS5350, Cisco AS5400, Cisco AS5850, Cisco AS5300, Cisco AS5800, and Cisco MC3810.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Each Real-Time Streaming Protocol (RTSP) session has a TCP port for control and a UDP (RTP) port for delivery of data. The control connection (TCP socket) is used to exchange a set of messages (request from the RTSP client and the response from the server) for displaying a prompt. The **debug rtsp socket** command enables the user to debug the message exchanges being done on the TCP control connection.

Note We recommend that you log output from the **debug rtsp socket** command to a buffer rather than sending the output to the console; otherwise, the size of the output could severely impact the performance of the gateway.

Related Commands

Command	Description
debug rtsp api	Displays debugging output for the RTSP client API.
debug rtsp error	Displays error message for RTSP data.
debug rtsp pmh	Displays debugging messages for the PMH.
voice call debug	Allows configuration of the voice call debugging output.

debug rudpv1

For debug information for Reliable User Datagram Protocol (RUDP), use the **debug rudpv1** command in privileged EXEC mode. To disable debugging output, use the **no**form of this command.

debug rudpv1 {**application** | **performance** | **retransmit** | **segment** | **signal** | **state** | **timer** | **transfer**}
no debug rudpv1 {**application** | **performance** | **retransmit** | **segment** | **signal** | **state** | **timer** | **transfer**}

Syntax Description

application	Application debugging.
performance	Performance debugging.
retransmit	Retransmit/soft reset debugging.
segment	Segment debugging.
signal	Signals sent to applications.
state	State transitions.
timer	Timer debugging.
transfer	Transfer state information.

Command Default

Debugging for rudpv1 is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(1)T	This command was introduced.
12.2(4)T	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco MC3810.
12.2(2)XB	This command was implemented on the Cisco AS5350 and Cisco AS5400 universal gateways.
12.2(2)XB1	This command was implemented on the Cisco AS5850 platform.
12.2(8)T	This command was implemented on Cisco IAD2420 series integrated access devices (IADs).
12.2(11)T	This command was implemented on the Cisco AS5350, Cisco AS5400, and Cisco AS5850 platforms.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command only during times of low traffic.

Examples

The following is sample output from the **debug rudpv1 application** command:

```
Router# debug rudpv1 application
Rudpv1:Turning application debugging on
*Jan  1 00:20:38.271:Send to appl (61F72B6C), seq 12
*Jan  1 00:20:48.271:Send to appl (61F72B6C), seq 13
*Jan  1 00:20:58.271:Send to appl (61F72B6C), seq 14
*Jan  1 00:21:08.271:Send to appl (61F72B6C), seq 15
*Jan  1 00:21:18.271:Send to appl (61F72B6C), seq 16
*Jan  1 00:21:28.271:Send to appl (61F72B6C), seq 17
*Jan  1 00:21:38.271:Send to appl (61F72B6C), seq 18
*Jan  1 00:21:48.275:Send to appl (61F72B6C), seq 19
*Jan  1 00:21:58.275:Send to appl (61F72B6C), seq 20
*Jan  1 00:22:08.275:Send to appl (61F72B6C), seq 21
*Jan  1 00:22:18.275:Send to appl (61F72B6C), seq 22
*Jan  1 00:22:28.275:Send to appl (61F72B6C), seq 23
*Jan  1 00:22:38.275:Send to appl (61F72B6C), seq 24
*Jan  1 00:22:48.279:Send to appl (61F72B6C), seq 25
*Jan  1 00:22:58.279:Send to appl (61F72B6C), seq 26
*Jan  1 00:23:08.279:Send to appl (61F72B6C), seq 27
*Jan  1 00:23:18.279:Send to appl (61F72B6C), seq 28
*Jan  1 00:23:28.279:Send to appl (61F72B6C), seq 29
```

The following is sample output from the **debug rudpv1 performance** command:

```
Router# debug rudpv1 performance
Rudpv1:Turning performance debugging on
corsair-f#
*Jan  1 00:44:27.299:
*Jan  1 00:44:27.299:Rudpv1 Sent:Pkts 11,  Data Bytes 236,  Data Pkts 9
*Jan  1 00:44:27.299:Rudpv1 Rcvd:Pkts 10,  Data Bytes 237,  Data Pkts 9
*Jan  1 00:44:27.299:Rudpv1 Discarded:0,  Retransmitted 0
*Jan  1 00:44:27.299:
*Jan  1 00:44:37.299:
*Jan  1 00:44:37.299:Rudpv1 Sent:Pkts 11,  Data Bytes 236,  Data Pkts 9
*Jan  1 00:44:37.299:Rudpv1 Rcvd:Pkts 10,  Data Bytes 237,  Data Pkts 9
*Jan  1 00:44:37.299:Rudpv1 Discarded:0,  Retransmitted 0
*Jan  1 00:44:37.299:
*Jan  1 00:44:47.299:
*Jan  1 00:44:47.299:Rudpv1 Sent:Pkts 11,  Data Bytes 236,  Data Pkts 9
*Jan  1 00:44:47.299:Rudpv1 Rcvd:Pkts 11,  Data Bytes 236,  Data Pkts 9
*Jan  1 00:44:47.299:Rudpv1 Discarded:0,  Retransmitted 0
*Jan  1 00:44:47.299:
```

The following is sample output from the **debug rudpv1 retransmit** command:

```
Router# debug rudpv1 retransmit
Rudpv1:Turning retransmit/softreset debugging on
*Jan  1 00:52:59.799:Retrans timer, set to ack 199
*Jan  1 00:52:59.903:Retrans timer, set to ack 200
*Jan  1 00:53:00.003:Retrans timer, set to ack 201
*Jan  1 00:53:00.103:Retrans timer, set to ack 202
*Jan  1 00:53:00.203:Retrans timer, set to ack 203
*Jan  1 00:53:00.419:Retrans timer, set to ack 97
*Jan  1 00:53:00.503:Retrans handler fired, 203
*Jan  1 00:53:00.503:Retrans:203:205:
*Jan  1 00:53:00.503:
*Jan  1 00:53:00.607:Retrans timer, set to ack 207
*Jan  1 00:53:00.907:Retrans timer, set to ack 210
*Jan  1 00:53:01.207:Retrans handler fired, 210
*Jan  1 00:53:01.207:Retrans:210:211:212:
*Jan  1 00:53:01.207:
*Jan  1 00:53:01.207:Retrans timer, set to ack 213
*Jan  1 00:53:01.311:Retrans timer, set to ack 214
```

```
*Jan  1 00:53:01.419:Retrans timer, set to ack 98
*Jan  1 00:53:01.611:Retrans timer, set to ack 215
*Jan  1 00:53:01.711:Retrans timer, set to ack 218
*Jan  1 00:53:01.811:Retrans timer, set to ack 219
*Jan  1 00:53:01.911:Retrans timer, set to ack 220
*Jan  1 00:53:02.011:Retrans timer, set to ack 221
*Jan  1 00:53:02.311:Retrans handler fired, 221
*Jan  1 00:53:02.311:Retrans:221:
*Jan  1 00:53:02.311:
*Jan  1 00:53:02.311:Retrans timer, set to ack 222
*Jan  1 00:53:02.415:Retrans timer, set to ack 225
```

The following is sample output from the **debug rudpv1 segment** command:

```
Router# debug rudpv1 segment
Rudpv1:Turning segment debugging on
*Jan  1 00:41:36.359:Rudpv1: (61F72DAC) Rcvd ACK 61..198 (32)
*Jan  1 00:41:36.359:Rudpv1: (61F72DAC) Send ACK 199..61 (32)
*Jan  1 00:41:36.459:Rudpv1: (61F72DAC) Rcvd ACK 62..199 (8)
*Jan  1 00:41:36.459:Rudpv1: (61F72DAC) Rcvd ACK 62..199 (32)
*Jan  1 00:41:36.459:Rudpv1: (61F72DAC) Send ACK 200..62 (32)
*Jan  1 00:41:36.559:Rudpv1: (61F72DAC) Rcvd ACK 63..200 (32)
*Jan  1 00:41:36.559:Rudpv1: (61F72DAC) Send ACK 201..63 (32)
*Jan  1 00:41:36.659:Rudpv1: (61F72DAC) Rcvd ACK 64..201 (32)
*Jan  1 00:41:36.659:Rudpv1: (61F72DAC) Send ACK 202..64 (32)
*Jan  1 00:41:36.759:Rudpv1: (61F72DAC) Rcvd ACK 65..202 (32)
*Jan  1 00:41:36.759:Rudpv1: (61F72DAC) Send ACK 203..65 (32)
*Jan  1 00:41:36.859:Rudpv1: (61F72DAC) Rcvd ACK 66..202 (32)
*Jan  1 00:41:36.859:Rudpv1: (61F72DAC) Send ACK 204..66 (32)
*Jan  1 00:41:36.959:Rudpv1: (61F72DAC) Rcvd ACK 67..202 (32)
*Jan  1 00:41:36.959:Rudpv1: (61F72DAC) Rcvd ACK EAK 68..202 (9)
*Jan  1 00:41:36.959:Rudpv1: (61F72DAC) Send ACK 203..67 (32)
*Jan  1 00:41:36.963:Rudpv1: (61F72DAC) Send ACK 205..67 (32)
*Jan  1 00:41:36.963:Rudpv1: (61F72DAC) Rcvd ACK 68..204 (8)
*Jan  1 00:41:37.051:Rudpv1: (61F72B6C) Send ACK NUL 118..96 (8)
*Jan  1 00:41:37.051:Rudpv1: (61F72B6C) Rcvd ACK 97..118 (8)
*Jan  1 00:41:37.059:Rudpv1: (61F72DAC) Rcvd ACK 68..205 (32)
*Jan  1 00:41:37.063:Rudpv1: (61F72DAC) Send ACK 206..68 (32)
*Jan  1 00:41:37.263:Rudpv1: (61F72DAC) Rcvd ACK 70..206 (32)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Send ACK EAK 207..68 (9)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Rcvd ACK 71..206 (32)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Rcvd ACK 69..206 (32)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Send ACK 207..71 (8)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Send ACK 207..71 (32)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Send ACK 208..71 (32)
*Jan  1 00:41:37.363:Rudpv1: (61F72DAC) Send ACK 209..71 (32)
*Jan  1 00:41:37.367:Rudpv1: (61F72DAC) Rcvd ACK 72..209 (8)
*Jan  1 00:41:37.463:Rudpv1: (61F72DAC) Rcvd ACK 72..209 (32)
*Jan  1 00:41:37.463:Rudpv1: (61F72DAC) Send ACK 210..72 (32)
*Jan  1 00:41:37.563:Rudpv1: (61F72DAC) Rcvd ACK 73..210 (32)
*Jan  1 00:41:37.563:Rudpv1: (61F72DAC) Send ACK 211..73 (32)
```

The following is sample output from the **debug rudpv1 signal** command:

```
Router# debug rudpv1 signal
Rudpv1:Turning signal debugging on
*Jan  1 00:39:59.551:Rudpv1:Sent CONN_FAILED to connID 61F72DAC, sess 33
*Jan  1 00:39:59.551:
*Jan  1 00:39:59.551:Rudpv1:Sent CONN_TRANS_STATE to connID 61F72B6C, sess 34
*Jan  1 00:39:59.551:
*Jan  1 00:39:59.551:Rudpv1:Sent CONN_TRANS_STATE to connID 61F72DAC, sess 33
*Jan  1 00:39:59.551:
*Jan  1 00:39:59.551:Rudpv1:Sent CONN_OPEN to connID 61F72B6C, sess 34
```

```
*Jan  1 00:39:59.551:Rudpv1:Sent AUTO_RESET to connID 61F72DAC, sess 33
*Jan  1 00:39:59.551:
*Jan  1 00:40:00.739:%LINK-5-CHANGED:Interface FastEthernet0, changed state
to administratively down
*Jan  1 00:40:01.739:%LINEPROTO-5-UPDOWN:Line protocol on Interface
FastEthernet0, changed state to down
*Jan  1 00:40:04.551:Rudpv1:Sent CONN_RESET to connID 61F72DAC, sess 33
*Jan  1 00:40:04.551:
*Jan  1 00:40:05.051:Rudpv1:Clearing conn rec values, index 2, connid
61F72DAC
*Jan  1 00:40:10.051:Rudpv1:Sent CONN_RESET to connID 61F72DAC, sess 33
*Jan  1 00:40:10.051:
*Jan  1 00:40:10.551:Rudpv1:Clearing conn rec values, index 2, connid
61F72DAC
*Jan  1 00:40:15.551:Rudpv1:Sent CONN_RESET to connID 61F72DAC, sess 33
*Jan  1 00:40:15.551:
*Jan  1 00:40:16.051:Rudpv1:Clearing conn rec values, index 2, connid
61F72DAC
*Jan  1 00:40:21.051:Rudpv1:Sent CONN_RESET to connID 61F72DAC, sess 33
*Jan  1 00:40:21.051:
*Jan  1 00:40:21.551:Rudpv1:Clearing conn rec values, index 2, connid
61F72DAC
*Jan  1 00:40:25.587:%LINK-3-UPDOWN:Interface FastEthernet0, changed state
to up
*Jan  1 00:40:26.551:Rudpv1:Sent CONN_RESET to connID 61F72DAC, sess 33
*Jan  1 00:40:26.551:
*Jan  1 00:40:26.587:%LINEPROTO-5-UPDOWN:Line protocol on Interface
FastEthernet0, changed state to up
*Jan  1 00:40:27.051:Rudpv1:Clearing conn rec values, index 2, connid
61F72DAC
*Jan  1 00:40:28.051:Rudpv1:Sent CONN_OPEN to connID 61F72DAC, sess 33
```

The following is sample output from the **debug rudpv1 state** command:

```
Router# debug rudpv1 state
Rudpv1:Turning state debugging on
*Jan  1 00:38:37.323:Rudpv1: (61F72DAC) State Change:OPEN -> CONN_FAILURE
*Jan  1 00:38:37.323:Rudpv1: (61F72B6C) State Change:OPEN -> TRANS_STATE
*Jan  1 00:38:37.323:Rudpv1: (61F72DAC) State Change:CONN_FAILURE ->
TRANS_STATE
*Jan  1 00:38:37.323:Rudpv1: (61F72B6C) State Change:TRANS_STATE -> OPEN
*Jan  1 00:38:37.323:Rudpv1: (61F72DAC) State Change:TRANS_STATE -> SYN_SENT
*Jan  1 00:38:37.455:%LINK-5-CHANGED:Interface FastEthernet0, changed state
to administratively down
*Jan  1 00:38:38.451:%LINEPROTO-5-UPDOWN:Line protocol on Interface
FastEthernet0, changed state to down
*Jan  1 00:38:42.323:Rudpv1: (61F72DAC) State Change:SYN_SENT -> CLOSED
*Jan  1 00:38:42.823:Rudpv1: (61F72DAC) State Change:INACTIVE -> SYN_SENT
*Jan  1 00:38:47.823:Rudpv1: (61F72DAC) State Change:SYN_SENT -> CLOSED
*Jan  1 00:38:48.323:Rudpv1: (61F72DAC) State Change:INACTIVE -> SYN_SENT
*Jan  1 00:38:53.323:Rudpv1: (61F72DAC) State Change:SYN_SENT -> CLOSED
*Jan  1 00:38:53.823:Rudpv1: (61F72DAC) State Change:INACTIVE -> SYN_SENT
*Jan  1 00:38:56.411:%LINK-3-UPDOWN:Interface FastEthernet0, changed state
to up
*Jan  1 00:38:57.411:%LINEPROTO-5-UPDOWN:Line protocol on Interface
FastEthernet0, changed state to up
*Jan  1 00:38:57.823:Rudpv1: (61F72DAC) State Change:SYN_SENT -> OPEN
```

The following is sample output from the **debug rudpv1 timer** command:

```
Router# debug rudpv1 timer
Rudpv1:Turning timer debugging on
*Jan  1 00:53:40.647:Starting Retrans timer for connP = 61F72B6C, delay = 300
```

```
*Jan  1 00:53:40.647:Stopping SentList timer for connP = 61F72B6C
*Jan  1 00:53:40.747:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:40.747:Stopping Retrans timer for connP = 61F72B6C
*Jan  1 00:53:40.747:Starting Retrans timer for connP = 61F72B6C, delay = 300
*Jan  1 00:53:40.747:Stopping SentList timer for connP = 61F72B6C
*Jan  1 00:53:40.847:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:40.847:Stopping Retrans timer for connP = 61F72B6C
*Jan  1 00:53:40.847:Starting Retrans timer for connP = 61F72B6C, delay = 300
*Jan  1 00:53:40.847:Stopping SentList timer for connP = 61F72B6C
*Jan  1 00:53:40.947:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:40.947:Stopping Retrans timer for connP = 61F72B6C
*Jan  1 00:53:40.947:Starting Retrans timer for connP = 61F72B6C, delay = 300
*Jan  1 00:53:40.947:Stopping SentList timer for connP = 61F72B6C
*Jan  1 00:53:41.047:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.147:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.151:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.151:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.151:Stopping Retrans timer for connP = 61F72B6C
*Jan  1 00:53:41.151:Starting SentList timer for connP = 61F72B6C, delay = 300
*Jan  1 00:53:41.419:Timer Keepalive (NullSeg) triggered for conn = 61F72DAC
*Jan  1 00:53:41.419:Starting Retrans timer for connP = 61F72DAC, delay = 300
*Jan  1 00:53:41.419:Stopping SentList timer for connP = 61F72DAC
*Jan  1 00:53:41.419:Starting NullSeg timer for connP = 61F72DAC, delay = 1000
*Jan  1 00:53:41.419:Stopping Retrans timer for connP = 61F72DAC
*Jan  1 00:53:41.451:Timer SentList triggered for conn = 61F72B6C
*Jan  1 00:53:41.451:Starting SentList timer for connP = 61F72B6C, delay = 300
*Jan  1 00:53:41.451:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.451:Stopping SentList timer for connP = 61F72B6C
*Jan  1 00:53:41.551:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.551:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.551:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
*Jan  1 00:53:41.551:Starting NullSeg timer for connP = 61F72B6C, delay = 1000
```

The following is sample output from the **debug rudpv1 transfer** command:

```
Router# debug rudpv1 transfer
Rudpv1:Turning transfer debugging on
*Jan  1 00:37:30.567:Rudpv1:Send TCS, connId 61F72B6C, old connId 61F72DAC
*Jan  1 00:37:30.567:Rudpv1:Initiate transfer state, old conn 61F72DAC to
new conn 61F72B6C
*Jan  1 00:37:30.567:Rudpv1:Old conn send window 51 .. 52
*Jan  1 00:37:30.567:Rudpv1:New conn send window 255 .. 2
*Jan  1 00:37:30.567:Rudpv1:Rcvd TCS 142, next seq 142
*Jan  1 00:37:30.567:Rudpv1:Rcv'ing trans state, old conn 61F72DAC to new
conn 61F72B6C
*Jan  1 00:37:30.567:Rudpv1:Seq adjust factor 148
*Jan  1 00:37:30.567:Rudpv1:New rcvCur 142
*Jan  1 00:37:30.567:Rudpv1:Send transfer state, old conn 61F72DAC to new
conn 61F72B6C
*Jan  1 00:37:30.567:Rudpv1:Send TCS, connId 61F72B6C, old connId 61F72DAC,
seq adjust 208, indication 0
*Jan  1 00:37:30.567:Rudpv1:Transfer seg 51 to seg 3 on new conn
*Jan  1 00:37:30.567:Rudpv1:Finishing transfer state, old conn 61F72DAC to
new conn 61F72B6C
*Jan  1 00:37:30.567:Rudpv1:Send window 2 .. 4
```

Related Commands

Command	Description
clear rudpv1 statistics	Clears RUDP statistics and failure counters.
show rudpv1	Displays RUDP failures, parameters, and statistics.