E through event manager Commands

E through event manager Commands

event application

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of an event raised through the EEM Event Publish application programming interface (API), use the **event applicationcommand** in applet configuration mode. To remove the application event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **application** **subsystem** *subsystem-id* **type** *event-type* [**maxrun** *maxruntime-number*]
no [**tag** *event-tag*] **event** **application** **subsystem** *subsystem-id* **type** *event-type* [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
2subsystem	Specifies an identifier for the subsystem that will publish the application event.
subsystem-id	Number in the range from 1 to 4294967295 that identifies the subsystem. When an event is to be published by an EEM policy, the *subsystem-id* reserved for a policy is 798.
type	Specifies an event type within the specified event.
event-type	Integer in the range from 1 to 4294967295.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM event criteria are specified.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(25)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.

Release	Modification
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.

Usage Guidelines

An EEM event is triggered when an application calls the EEM Event Publish API with an event specification that matches the subsystem ID and application event type.

Examples

The following example shows how a policy named EventPublish_A runs every 20 seconds and publishes an event to a well-known EEM event type numbered 1. A second policy named EventPublish_B is registered to run when the well-known EEM event type of 1 occurs. When policy EventPublish_B runs, it outputs a message to syslog containing data passed as an argument from EventPublish_A.

```
Router(config)# event manager applet EventPublish_A
Router(config-applet)# event timer watchdog time 20.0
Router(config-applet)# action 1.0 syslog msg "Applet EventPublish_A"
Router(config-applet)# action 2.0 publish-event sub-system 798 type 1 arg1 twenty
Router(config-applet)# exit
Router(config)# event manager applet EventPublish_B
Router(config-applet)# event application subsystem 798 type 1
Router(config-applet)# action 1.0 syslog msg "Applet EventPublish_B arg1
$_application_data1"
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event cli

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run by matching a Cisco IOS command-line interface (CLI) command, use the **event cli**command in applet configuration mode. To remove the CLI command event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **cli** **pattern** *regular-expression* [**default**] [**enter**] [**questionmark**] [**tab**] [{**sync** {**yes** | **no**} | **skip** {**yes** | **no**}}] [**mode** *variable*] [**occurs** *num-occurrences*] [**period** *period-value*] [**maxrun** *maxruntime-number*]
no **event** [**tag** *event-tag*] **cli**

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
pattern	Specifies the regular expression used to perform the CLI command pattern match. The CLI command must have been successfully parsed before the pattern match is attempted. The pattern match is compared with the fully expanded CLI command string.
regular-expression	Regular expression. If the expression contains embedded blanks, enclose it in double quotation marks.
default	(Optional) The time period during which the CLI event detector waits for the policy to exit (specified in ssssssssss[.mmm] format, where ssssssssss must be an integer representing seconds from 0 to 4294967295, and where mmm must be an integer representing milliseconds from 0 to 999). If the default time period expires before the policy exits, the default action will be executed. The default action is to run the command. If this argument is not specified, the default time period is set to 30 seconds.
enter	Specifies the event match when the user presses the Enter key.
questionmark	Specifies the event match when the user presses the Questionmark key.
tab	Specifies the event match when the user presses the Tab key.
sync	Indicates whether the policy should be executed synchronously before the CLI command executes. • If the **yes** keyword is specified, the policy will run synchronously with the CLI command. • If the **no** keyword is specified, the policy will run asynchronously with the CLI command.
skip	Indicates whether the CLI command should be executed. This keyword is required if the **sync** keyword is followed by the **no** keyword. If the **sync** keyword is followed by the **yes** keyword, the **skip** keyword should not be specified. • If the **yes** keyword is specified, the CLI command will not be executed. • If the **no** keyword is specified, the CLI command will be executed. This is the default. **Caution** When the **skip** keyword is followed by the **yes** keyword, unintended results may be produced if the pattern match is made for configuration commands because the CLI command that matches the regular expression will not be executed.
mode *variable*	Specifies the CLI parser mode events for the keywords that follow.
occurs	(Optional) Specifies the number of matching occurrences before an EEM event is triggered. When a number is not specified, an EEM event is triggered after the first match.

num-occurrences	(Optional) Integer greater than 0 that specifies the number of occurrences.
period	(Optional) Specifies a backward looking time window in which all CLI events must occur (the occurs clause must be satisfied) in order for an event to be published (specified in SSSSSSSSSS[.MMM] format, where SSSSSSSSSS must be an integer representing seconds between 0 and 4294967295, inclusive, and where MMM must be an integer representing milliseconds between 0 and 999). If this argument is not specified, the most recent event is used.
period-value	(Optional) Integer that represents seconds and optional milliseconds in the format ssssssssss[.mmm]. Seconds is an integer in the range from 0 to 4294967295. Milliseconds is an integer in the range from 0 to 999. When you specify milliseconds only, use the format 0.mmm.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.

Command Default

No EEM events are triggered on the basis of a match with a Cisco IOS CLI command.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.
12.4(22)T	The **default**, **enter**, **mode, questionmark**, and **tab** keywords were added to support CLI parser-based events.

Usage Guidelines

Use the **event cli** command to set up event criteria against which CLI commands are matched. CLI commands are compared against a specified regular expression. After a specified number of matches occurs within a specified time period, an EEM event is triggered. If multiple conditions exist, the EEM event is triggered when all the conditions are met.

When the **sync**keyword is used, the event detector is notified when the policy completes running. The exit status of the policy determines whether the CLI command will be executed. If the policy exit status is zero--the policy ran successfully--the CLI command is not executed; otherwise the CLI command runs.

The EEM applet can accept four keywords to add CLI parser-based events. The behavior of these keywords are as follows:

- The **default** keyword is used to perform the action during which the CLI event detector waits for the policy to exit.
- The **tab** keyword is used to expand abbreviated commands or parameters. A new prompt line is displayed with the expanded text.
- The **questionmark** keyword is used to display a list with help of valid commands or parameters. Help is displayed first followed by a new prompt line.
- The **enter** keyword will parse and run the command.

Regular Expression Match

The CLI event detector screens CLI commands for a regular expression match. When a match is found, an event is published. The match logic is performed on the fully expanded CLI command after the command is successfully parsed and before it is executed. The CLI event detector supports three publish modes:

- Synchronous publishing of CLI events--The CLI command is not executed until the EEM policy exits, and the EEM policy can control whether the command is executed. The read/write variable, _exit_status, allows you to set the exit status at policy exit for policies triggered from synchronous events. If _exit_status is 0, the command is skipped, if _exit_status is 1, the command is run.
- Asynchronous publishing of CLI events--The CLI event is published, and then the CLI command is executed.
- Asynchronous publishing of CLI events with command skipping--The CLI event is published, but the CLI command is not executed.

Examples

The following configuration will prevent you to access the configuration mode. If the _exit_status is not set to 1, the EEM policy will disable functionality in the router. To remove this configuration, you may need to reload with unsaved configuration. It is possible to remove the event applet but all unsaved configuration will be lost and a complete reconfiguration will be necessary.

```
event manager applet test_cli1
  event cli pattern “config” sync yes
  action 1.0 syslog msg “Config command is entered”
end
```

Caution Failure to set the exit_status to 1 will cause the default action which is to skip executing the CLI command. This can lead to situations where the router has to be reloaded in order to continue operations.

The following example shows how to specify an EEM applet to run when the Cisco IOS **write memory** CLI command is run. The applet provides a notification via a syslog message that this event has occurred. The _exit_status is set to 1.

```
Router(config)# event manager applet cli-match
Router(config-applet)# event cli pattern "write memory.*" sync yes
Router(config-applet)# action 1.0 syslog msg "$_cli_msg Command Executed"
Router(config-applet)# set 2.0 _exit_status 1
```

The following example shows how unintended results can be produced when using the **skip** keyword followed by the **yes** keyword. When the **skip** keyword is followed by the **yes**keyword, unintended results may be produced if the pattern match is made for configuration commands because the CLI command that matches the regular expression will not be executed. In this example, the first applet (ap1) uses the **skip** keyword followed by the **yes** keyword to specify that any CLI command that contains the **show ip interface**pattern is not executed. This results in the second applet (ap2) being configured without an event statement because it contains the **show ip interface** pattern.

```
Router(config)# event manager applet ap1
Router(config-applet)# event cli pattern "show ip interface" sync no skip yes occurs 1
period 5
Router(config-applet)# action 1 syslog msg "test 1"
Router(config-applet)# exit
Router(config)# event manager applet ap2
Router(config-applet)# event cli pattern "show ip interface" sync no skip no occurs 1
period 5
Router(config-applet)# action 1 syslog msg "test 2"
Router(config-applet)# end
```

The following example shows how CLI parser-based events can be generated:

```
Router(config)# event manager applet ap1
Router(config-applet)# event cli pattern "show ip interface"
```

The results are displayed on the screen. Note that the second line contains a message that no event is configured for the EEM applet ap2. Use command CLI pattern matching with caution when the **skip** and **yes** keywords are specified.

```
00:00:41: %HA_EM-6-LOG: ap1: test 1
00:00:41: %HA_EM-4-FMPD_NO_EVENT: No event configured for applet ap2
router#show run | beg event event manager applet ap1 event cli pattern "show ip
interface" sync no skip yes occurs 1 period 5 action 1 syslog msg "test 1"
event manager applet ap2
 action 1 syslog msg "test 2"
!
end
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

event counter

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of a named counter crossing a threshold, use the **event counter**command in applet configuration mode. To remove the counter event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **counter** **name** *counter-name* **entry-op** *operator* **entry-val** *entry-value* [**exit-op** *operator*] [**exit-val** *exit-value*] [**maxrun** *maxruntime-number*]
no **event** [**tag** *event-tag*] **counter** **name** *counter-name* **entry-op** *operator* **entry-val** *entry-value* [**exit-op** *operator*] [**exit-val** *exit-value*] [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
name	Specifies that a counter will be monitored.
counter-name	Name of the counter that will be monitored.
entry-op	Compares the contents of the current counter value with the entry value using a specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.
operator	Value used with the **entry-op** and **exit-op** keywords that determines how the current counter value is compared to the entry value or the exit value. Valid values are: • **gt** --Greater than. • **ge** --Greater than or equal to. • **eq** --Equal to. • **ne** --Not equal to. • **lt** --Less than. • **le** --Less than or equal to.
entry-val	Specifies the value with which the contents of the current counter are compared to decide if a counter event should be raised.
entry-value	Number in the range from -2147483648 to 2147483647, inclusive.
exit-op	(Optional) Compares the contents of the current counter with the exit value using a specified operator. If there is a match, an event is triggered and event monitoring is reenabled.
exit-val	(Optional) Specifies the value with which the contents of the current counter are compared to decide whether the exit criteria are met.
exit-value	(Optional) Number in the range from -2147483648 to 2147483647, inclusive.

maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss [.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM events are triggered on the basis of a named counter crossing a threshold.

Command Modes

Event counter applet configuration (config-applet-event-counter)

Command History

Release	Modification
12.2(25)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(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.

Usage Guidelines

An EEM event is triggered when the value of a specified counter crosses a defined threshold. Depending on the operator, the threshold may be crossed when the value is greater than the threshold or when the value is less than the threshold.

Use the **event counter** command with the **action counter** command when an event occurs periodically and you want an action to be implemented after a specified number of occurrences of the event.

Exit criteria are optional. If exit criteria are not specified, event monitoring will be reenabled immediately. If exit criteria are specified, event monitoring is not reenabled until the criteria are met.

Examples

The following example shows that policy EventCounter_A is configured to run once a minute and to increment a well-known counter called critical_errors. A second policy--EventCounter_B--is registered to be triggered when the well-known counter called critical_errors exceeds a threshold of 3. When policy EventCounter_B runs, it resets the counter to 0.

```
Router(config)# event manager applet EventCounter_A
Router(config-applet)# event timer watchdog time 60.0
Router(config-applet)# action 1.0 syslog msg "EventCounter_A"
Router(config-applet)# action 2.0 counter name critical_errors value 1 op inc
Router(config-applet)# exit
Router(config)# event manager applet EventCounter_B
Router(config-applet)# event counter name critical_errors entry-op gt entry-val 3 exit-op
lt exit-val 3
Router(config-applet)# action 1.0 syslog msg “EventCounter_B”
Router(config-applet)# action 2.0 counter name critical_errors value 0 op set
```

Related Commands

Command	Description
action counter	Sets or modifies a named counter when an Embedded Event Manager applet is triggered.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event gold

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of a Generic Online Diagnostic (GOLD) failure event when monitoring one or more cards and optional subcards, use the **event gold** command in applet configuration mode. To remove the report event criteria, use the **no** form of this command.

event gold card {**all***card-number*} [**subcard** {**all***subcard-number*}] [**new-failure** {**true** | **false**}] [**severity-major**] [**severity-minor**] [**severity-normal**] [**action-notify** {**true** | **false**}] [**testing-type** {**bootup** | **ondemand** | **schedule** | **monitoring**}] [**test-name** *test-name*] [**test-id** *test-id*] [**consecutive-failure** *consecutive-failure-number*] [**platform-action** *action-flag-number*] [**maxrun** *maxruntime-number*]
no event gold card {**all***card-number*} [**subcard** {**all***subcard-number*}] [**new-failure** {**true** | **false**}] [**severity-major**] [**severity-minor**] [**severity-normal**] [**action-notify** {**true** | **false**}] [**testing-type** {**bootup** | **ondemand** | **schedule** | **monitoring**}] [**test-name** *test-name*] [**test-id** *test-id*] [**consecutive-failure** *consecutive-failure-number*] [**platform-action** *action-flag-number*] [**maxrun** *maxruntime-number*]

Syntax Description

card	Specifies that all or one card must be monitored. Either **all** or *card-number* must be specified. • **all** --Specifies that all cards are to be monitored. This is the default. • *card-number* --Number of a specific card to be monitored. **Note** The **card** keyword is required to complete the **event gold**command.
subcard	(Optional) Specifies that one or more subcards are to be monitored. If the **subcard** keyword is specified, then **all** or *subcard-number*value must be specified. • **all** --Specifies that all subcards are to be monitored. • *subcard-number*-- Number of a subcard to be monitored. If the **subcard** keyword is not specified, the default is **all**.

new-failure	(Optional) Specifies event criteria based on the new test failure information from GOLD. If the **new-failure** keyword is specified, then the **true** or **false** keyword must be specified. • **true** --Specifies that the event criteria for the new test failure is true from GOLD. • **false** --Specifies that the event criteria for the new test failure is false from GOLD. If the **new-failure** keyword is not specified, the new test failure information from GOLD is not considered in the event criteria.
severity-major	(Optional) Specifies that the event criteria for diagnostic result matches with diagnostic major error from GOLD.
severity-minor	(Optional) Specifies that the event criteria for diagnostic result matches with diagnostic minor error from GOLD.
severity-normal	(Optional) Specifies that the event criteria for diagnostic result matches with diagnostic normal from GOLD. This is the default.
action-notify	(Optional) Specifies the event criteria based on the action notify information from GOLD. If the **action-notify** keyword is specified, then **true** or **false** keyword must be specified. • **true** --Specifies that the event criteria for the action notify is true from GOLD. • **false** --Specifies that the event criteria for the action notify is false from GOLD. If the **action-notify** keyword is not specified, the action notify information from GOLD is not considered in the event criteria.
testing-type	(Optional) Specifies the event criteria based on the testing types of diagnostic from GOLD. If the **testing-type** keyword is specified, then bootup, ondemand, schedule, or monitoring must be specified. • **bootup** --Specifies the diagnostic tests running on system bootup. • **ondemand** --Specifies the diagnostic tests running from CLI after the card is online. • **schedule** --Specifies the scheduled diagnostic tests. • **monitoring** --Specifies the diagnostic tests that are running periodically in the background to monitor the health of the system. If the **testing-type** keyword is not specified, the testing type information from GOLD is not considered in the event criteria and the policy applies to all the diagnostic testing types.
test-name	(Optional) Specifies the event criteria based on the test name. If the **test-name** keyword is specified, then the *test-name*value must be specified. • *test-name* --Name of the test. If the **test-name** keyword is not specified, the test name information from GOLD is not considered in the event criteria.

test-id	(Optional) Specifies the event criteria based on test ID. Because the test ID can be different for the same test on different line cards, usually the **test-name** keyword should be used instead. If the test ID is specified and has conflicts with the specified test name, the test name overwrites the test ID. If the **test-id** keyword is specified, the *test-id*value must be specified. • *test-id*-- ID number of the test. The limit is 65535. If the **test-id** keyword is not specified, test ID information from GOLD is not considered in the event criteria.
consecutive-failure	(Optional) Specifies the event criteria based on consecutive test failure information from GOLD. If the **consecutive-failure** keyword is specified, the c*onsecutive-failure-number*value must be specified. • *consecutive-failure-number*-- Number of consecutive failures. If the **consecutive-failure** keyword is not specified, consecutive test failure information from GOLD is not considered in the event criteria.
platform-action	(Optional) Specifies whether callback to the platform is needed when all the event criteria are matched. When callback is needed, the platform needs to register a callback function through the provided registry. If the **platform-action** keyword is specified, the *action-flag-number* value must be specified. • *action-flag-number*-- Number of the action flag that provides the platform with more specific information when callback is performed. The action flag is platform specific. It is up to the platform to determine what action needs to be taken based on the flag. The maximum number is 65535. If the **platform-action** keyword is not specified, there is no callback.
maxrun	(Optional) Specifies the maximum runtime of the script. If the **maxrun**keyword is specified, the *maxruntime-number*value must be specified. • *maxruntime-number*-- Maximum runtime number in seconds. The maximum number is 4294967295 seconds. If the **maxrun**keyword is not specified, the default runtime is 20 seconds.

Command Default

No EEM event criteria are specified.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(18)SXF2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	The **action-notify**, **testing-type**, **test-name**, **test-id**, **consecutive-failure**, **platform-action**, and the **maxrun**keywords were added.

Release	Modification
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.

Usage Guidelines

You must enter the **event gold** command with the mandatory keyword **card**. For example, enter **event gold c ard** specifying either the **all** keyword or the *card-number* attribute; otherwise the command is incomplete. All other keywords are optional; however, once an optional keyword is specified, for example **new-failure**, its corresponding **true** or **false keyword must**be specified (the value is not optional anymore). The same principle is applicable for all other keywords that have specific values.

Examples

The following example shows how to specify that an EEM applet runs when a new GOLD failure event occurs for any card and any subcard. The applet sends a message to the CNS Event Bus to state that a GOLD failure event has occurred.

```
Router(config)# event manager applet gold-match
Router(config-applet)# event gold card all subcard all new-failure true
Router(config-applet)# action 1.0 cns-event msg "A GOLD failure event has occurred”
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event identity

To publish an event after authentication, authorization or normal traffic has begun to flow on the interface, use the **event identity**command in applet configuration mode. To disable the publishing of events, use the **no** form of this command.

event [**tag** *event-tag*] **identity interface** {*type number* | **regexp** *interface-name*} [**maxrun** *maxruntime-number*] [**aaa-attribute** *attribute-name*] [**authc** {**all** | **fail** | **success**}] [**authz** {**all** | **fail** | **success**}] [**authc-complete**] [**mac-address** *mac-address*]
no event identity

Syntax Description

tag	(Optional) Specifies a tag using the event-tag argument that can be used with the trigger command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
interface	Specifies the interface.
type number	Interface type and number.
regexp *interface-name*	Specifies a regular expression pattern to match against interface names.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the maxrun keyword is specified, the maxruntime-number value must be specified. If the maxrun keyword is not specified, the default applet run time is 20 seconds.

maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.
aaa-attribute	(Optional) Specifies the regular expression pattern for AAA attributes.
attribute-name	(Optional) AAA attribute name.
authc	(Optional) Triggers events on successful, failed or both successful and failed authentication. You must specify one of the following: • **all** --Triggers an event in all cases of authentication. • **fail** --Triggers an event if authentication fails. • **success** --Triggers an event if authentication is successful.
authz	(Optional) Trigger events on successful, failed or both successful and failed authorization. You must specify one of the following: • **all** --Triggers an event in all cases of authorization. • **fail** --Triggers an event if authorization fails. • **success** --Triggers an event if authorization is successful.
authz-complete	(Optional) Triggers events once the device connected to the interface is fully authenticated, authorized and normal traffic has begun to flow on that interface.
mac-address	(Optional) Specifies the MAC address.
mac-address	(Optional) The MAC address.

Command Default

By default, no events are published.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(52)SE	This command was introduced.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines

You must specify an interface. You can specify any or all of the other keywords. The keywords can be used in any combination.

Examples

The following example shows how to publish an event when authorization is successful or failure and when the device connected to the interface is fully authenticated, authorized and normal traffic has begun to flow on that interface:

```
Router(config)# event manager applet identity
Router(config-applet)# event identity interface fastethernet0 authz all athuz-complete
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event interface

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of a generic interface counter crossing a threshold or reaching exit criteria, use the **event interface**command in applet configuration mode. To remove the interface event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **interface name** *interface-type interface-number* **parameter** *counter-name* **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value** | **increment** | **rate**} **poll-interval** *poll-int-value* [**exit-comb** {**or** | **and**}] [**exit-op** *operator* **exit-val** *exit-value*] [**exit-type** {**value** | **increment** | **rate**}] [**exit-time** *exit-time-value*] [**average-factor** *average-factor-value*] [**maxrun** *maxruntime-number*]
no event [**tag** *event-tag*] **interface name** *interface-type interface-number* **parameter** *counter-name* **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value** | **increment** | **rate**} **poll-interval** *poll-int-value* [**exit-comb** {**or** | **and**}] [**exit-op** *operator* **exit-val** *exit-value*] [**exit-type** {**value** | **increment** | **rate**}] [**exit-time** *exit-time-value*] [**average-factor** *average-factor-value*] [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
name	Specifies the type and number of the interface to monitor.
interface-type	String that identifies the type of interface.
interface-number	Integer value that identifies the interface.
parameter	Specifies the name of the counter to monitor.

counter-name	Name of the counter. Supported values for the *counter-name* argument are the following: • **input_errors** --Includes runts, giants, no buffer, cyclic redundancy checksum (CRC), frame, overrun, and ignored counts. Other input-related errors can also cause the input errors count to be increased. Some datagrams may have more than one error. • **input_errors_crc** --Number of packets with a CRC generated by the originating LAN station or remote device that do not match the checksum calculated from the data received. • **input_errors_frame** --Number of packets received incorrectly that have a CRC error and a noninteger number of octets. • **input_errors_overrun** --Number of times the receiver hardware was unable to hand over received data to a hardware buffer because the input rate exceeded the receiver’s ability to handle the data. • **input_packets_dropped** --Number of packets dropped because of a full input queue. • **interface_resets** --Number of times an interface has been completely reset.

	• **output_buffer_failures** --Number of failed buffers and number of buffers swapped out. • **output_buffer_swappedout** --Number of packets swapped to Dynamic RAM (DRAM). • **output_errors** --Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. This may not balance with the sum of the output errors because some datagrams may have more than one error and other datagrams may have errors that do not fall into any of the specifically tabulated categories. • **output_errors_underrun** --Number of times the transmitter has been running faster than the router can handle. • **output_packets_dropped** --Number of packets dropped because of a full output queue. • **receive_broadcasts** --Number of broadcast or multicast packets received by the interface. • **receive_giants** --Number of packets that are discarded because they exceed the maximum packet size of the medium. • **receive_rate_bps** --Interface receive rate, in bytes per second. • **receive_rate_pps** --Interface receive rate, in packets per second. • **receive_runts** --Number of packets that are discarded because they are smaller than the minimum packet size of the medium. • **receive_throttle** --Number of times the receiver on the port was disabled, possibly because of buffer or processor overload. • **reliability** --Reliability of the interface, as a fraction of 255 (255 out of 255 is 100 percent reliability), calculated as an exponential average over 5 minutes. • **rxload** --Receive rate of the interface, as a fraction of 255 (255 out of 255 is 100 percent). • **transmit_rate_bps** --Interface transmit rate, in bytes per second. • **transmit_rate_pps** --Interface transmit rate, in packets per second. • **txload** --Transmit rate of the interface, as a fraction of 255 (255 out of 255 is 100 percent).
entry-op	Compares the current interface counter value with the entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.

operator	Value used with the **entry-op**and **exit-op** keywords that determines how the current counter value is compared with the entry value or the exit value. Valid values are: • **gt** --Greater than • **ge** --Greater than or equal to • **eq** --Equal to • **ne** --Not equal to • **lt** --Less than • **le** --Less than or equal to
entry-val *entry-value*	Specifies the value with which the current interface counter value is compared to decide if the interface event should be raised. Range is from -2147483648 to 2147483647.
entry-type	Specifies a type of operation to be applied to the object ID specified by the *entry-value* argument.
value	Value is defined as the actual value of the *entry-value* or *exit-value*argument.
increment	Increment uses the *entry-value* or *exit-value*field as an incremental difference. The *entry-value*or *exit-value*is compared with the difference between the current counter value and the value when the event was last triggered (or the first polled sample if this is a new event). A negative value checks the incremental difference for a counter that is decreasing.
rate	Rate is defined as the average rate of change over a period of time. The time period is the *average-factor-value* multiplied by the *poll-int-value*. At each poll interval the difference between the current sample and the previous sample is taken and recorded as an absolute value. An average of the previous *average-factor-value* samples is taken to be the rate of change.
poll-interval	Specifies the time interval between consecutive polls. The default is 1 second.
poll-int-value	Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 60 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds, specify the milliseconds in the format s.mmm. The poll interval value must not be less than 1 second. The default is 1 second.
exit-comb	(Optional) Indicates the combination of exit conditions that must be met before event monitoring is reenabled.
or	Indicates that both **exit-op** or **exit-val** and **exit-time**values must exist
and	Indicates that either **exit-op** or **exit-val** or **exit-time** values must exist
exit-op	(Optional) Compares the contents of the current interface counter value with the exit value using the specified operator. If there is a match, an event is triggered and event monitoring is reenabled.

exit-val *exit-value*	(Optional) Specifies the value with which the contents of the current interface counter value are compared to decide whether the exit criteria are met. If an exit value is specified, you must configure an exit operator. Range is from -2147483648 to 2147483647.
exit-type	(Optional) Specifies a type of operation to be applied to the object ID specified by the *exit-value* argument.
exit-time	(Optional) Specifies the time period after which the event monitoring is reenabled. The timing starts after the event is triggered.
exit-time-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
average-factor	(Optional) Specifies a number used to calculate the period used for rate-based calculations. The *average-factor-value* is multiplied by the *poll-int-value* to derive the period in milliseconds.
average-factor-value	(Optional) Number in the range from 1 to 64. The minimum average factor value is 1.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.

Command Default

No EEM events are triggered on the basis of a generic interface counter crossing a threshold or reaching exit criteria.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(25)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(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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 modified. The **tag**, **entry-type**, **value**, **increment**, **rate**, **exit-type**, **average-factor**, and **maxrun**keywords and associated arguments were added. The **entry-val-is-increment**, **true**, **false**,and **exit-val-is-increment** keywords were removed.

Usage Guidelines

An EEM event is triggered when one of the fields specified by an interface counter crosses a defined threshold.

Note While registering a policy, an interface can be configured using this command without being physically present in the device but EEM does not begin any monitoring activity until the interface is physically present.

Exit criteria are optional. If you do not specify the exit criteria, event monitoring will be reenabled immediately. If you specify the exit criteria, on the basis of values or time periods, event monitoring is not reenabled until the exit criteria are met.

When you use the **exit-comb** keyword, the following criteria must be met:

- If you specify the **or** operator, an exit comparison operator and an exit object ID value, or an exit time value must exist.
- If you specify the **and** operator, an exit comparison operator, an exit object ID value, and an exit time value must exist.

Cisco IOS Releases 12.4(15)T, 12.2(33)SB, 12.2(33)SRA, and 12.2(33)SXH, and Prior Releases

The **entry-val-is-increment**keyword triggers one of the following actions:

- If you specify the **true** keyword, the *entry-value* is an increment and the interface event is raised whenever the incremental value is reached.
- If you specify the **false** keyword, the *entry-value* is an actual value and the interface event is raised whenever the actual value occurs. This is the default.

When the optional **exit-val-is-increment** keyword is used, the following occurs:

- If you specify the **true** keyword, the *exit-value* is an increment value and the event monitoring is reenabled whenever the incremental value is reached.
- If you specify the **false** keyword, the *exit-value* is an actual value and the event monitoring is reenabled whenever the actual value occurs. This is the default.

Cisco IOS Release 12.4(20)T and Later Releases

The **entry-type**keyword triggers one of the following actions:

- If you specify the **value** keyword, the *entry-value* is an actual value and the interface event is raised whenever the actual value occurs.
- If you specify the **increment** keyword, the *entry-value* is an increment and the interface event is raised whenever the incremental value is reached.
- If you specify the **rate** keyword, the *entry-value* is a rate of change and the interface event is raised whenever the rate of change value is reached.

When you use the optional **exit-type**keyword, the following occurs:

- If you specify the **value** keyword, the *exit-value* is an actual value and the event monitoring is reenabled whenever the actual value occurs. This is the default.
- If you specify the **increment** keyword, the *exit-value* is an increment and the event monitoring is reenabled whenever the incremental value is reached.
- If you specify the **rate** keyword, the *exit-value* is a rate of change and the event monitoring is reenabled whenever the rate of change value is reached.

Examples

The following example shows how a policy named EventInterface is triggered every time the receive_throttle counter for the FastEthernet interface 0/0 is incremented by 5. The polling interval to check the counter is specified to run once every 90 seconds.

```
Router(config)# event manager applet EventInterface
Router(config-applet)# event interface name FastEthernet0/0 parameter receive_throttle
entry-op ge entry-val 5 entry-val-is-increment true poll-interval 90
Router(config-applet)# action 1.0 syslog msg "Applet EventInterface"
```

The following example shows how a policy named EventInterface_Load is triggered every time the receive_rate_bps counter for the FastEthernet interface 0/0 reaches a rate of change of 10,000 with an average factor of 10. The polling interval to check the counter is specified to run once every 120 seconds. This example is for a Cisco IOS Release 12.4(20)T or later image.

```
Router(config)# event manager applet EventInterface_Load
Router(config-applet)# event interface name FastEthernet0/0 parameter receive_rate_bps
entry-op ge entry-val 10000 entry-type rate poll-interval 120 average-factor 10
Router(config-applet)# action 1.0 syslog msg "Applet EventInterface_Load"
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event ioswdsysmon

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of Cisco IOS system monitor counters crossing a threshold, use the **event ioswdsysmon**command in applet configuration mode. To remove the event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **ioswdsysmon sub1** *subevent1* [**timewin** *timewin-value*] [**sub12-op** {**and** | **or**} **sub2** *subevent2*] [**maxrun** *maxruntime-number*]
no [**tag** *event-tag*] **event ioswdsysmon sub1** *subevent1* [**timewin** *timewin-value*] [**sub12-op** {**and** | **or**} **sub2** *subevent2*] [**maxrun** *maxruntime-number*]

Subevent Syntax (for the subevent1 and subevent2 Arguments) for Cisco IOS Images
cpu-proc taskname *task-name* **op** *operator* **val** *value* [**period** *period-value*]
mem-proc taskname *task-name* **op** *operator* **val** *value* [**is-percent** {**true** | **false**}] [**period** *period-value*]

Subevent Syntax (for the subevent1 and subevent2 Arguments) for Cisco IOS Software Modularity Images

cpu-proc taskname *task-name* **path** *pid* **op** *operator* **val** *value* [**period** *period-value*]

mem-proc taskname *task-name* **path** *pid* **op** *operator* **val** *value* [**is-percent** {**true** | **false**}] [**period** *period-value*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
sub1	Specifies the first subevent.
subevent1	First subevent. Use the syntax shown under the Subevent Syntax heading.
timewin	(Optional) Specifies the time window within which all the subevents must occur for an event to be generated.
timewin-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
sub12-op	(Optional) Indicates the combination operator for comparison between subevent 1 and subevent 2.
and	(Optional) Specifies that the results of both subevent 1 and subevent 2 must cross the specified thresholds.
or	(Optional) Specifies that the results of either subevent 1 or subevent 2 must cross the specified thresholds.
sub2	(Optional) Specifies the second subevent.
subevent2	(Optional) Second subevent. Use the syntax shown under the Subevent Syntax heading.
Subevent Syntax	Specifies the use of a sample collection of CPU statistics.
cpu-proc	
mem-proc	Specifies the use of a sample collection of memory statistics.
taskname	Specifies a Cisco IOS task name. **Note** In Cisco IOS Release 12.2(18)SXF4 and later releases, Software Modularity images contain POSIX processes, and Cisco IOS processes were renamed as tasks.
task-name	Name of the Cisco IOS task to be monitored. If the value of the *task-name* argument contains embedded blanks, enclose it in double quotation marks.

path	(Supported only in Software Modularity images) Specifies a Cisco IOS Software Modularity path and process name. **Note** In Cisco IOS Release 12.2(18)SXF4 and later releases, Software Modularity images contain POSIX processes, and Cisco IOS processes were renamed as tasks.
pid	(Supported only in Software Modularity images) Process ID of the Software Modularity process to be monitored.
op	Compares the collected CPU or memory usage sample with the value specified in the *value* argument.
operator	Two-character string. The *operator*argument takes one of the following values: • **gt** --Greater than • **ge** --Greater than or equal to • **eq** --Equal to • **ne** --Not equal to • **lt** --Less than • **le** --Less than or equal to
val	Specifies the value with which the collected CPU or memory usage sample is compared to decide if the subevent should be raised.
value	Number in the range from 1 to 4294967295.
period	(Optional) Specifies the elapsed time period for the collection samples to be averaged.
period-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If only milliseconds are used, the format is 0.mmm. If the time period is 0, the most recent sample is used.
is-percent	(Optional) Indicates whether the *value* argument is a percentage.
true	(Optional) Specifies that the *value* argument is a percentage.
false	(Optional) Specifies that the *value* argument is not a percentage.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM events are triggered on the basis of Cisco IOS system monitor counters.

Command Modes

Applet configuration (config-applet).

Command History

Release	Modification
12.2(25)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(18)SXF4	The **path** keyword and *pid* argument were added and this command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5
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	The tag and maxrun keywords were added to support multiple event statements within an applet.

Usage Guidelines

An EEM event is triggered when one of the Cisco IOS system monitor counters crosses a defined threshold. Depending on the operator, the threshold may be crossed when the value exceeds the threshold or when the value is less than the threshold.

If a match is found when the **op**keyword is used, a subevent is triggered.

Examples

The following example shows how to configure a policy to trigger an applet when the total amount of memory used by the process named “IP RIB Update” has increased by more than 50 percent over the sample period of 60 seconds:

```
Router(config)# event manager applet IOSWD_Sample3
Router(config-applet)# event ioswdsysmon sub1 mem-proc taskname "IP RIB Update" op gt val
50 is-percent true period 60
Router(config-applet)# action 1 syslog msg "IOSWD_Sample3 Policy Triggered"
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event ipsla

To publish an event when an IP SLAs operation is triggered for an Embedded Event Manager (EEM) applet, use the **event ipsla** command in applet configuration mode. To disable publishing events when an IP SLAs reaction gets triggered, use the **no** form of this command.

event [**tag** *event-tag*] **ipsla** {**group-name** *name* [**operation-id** *operation-id-value*] | **operation-id** *operation-id-value* [**group-name** *name*]} [**dest-ip-address** *ip-address*] [**reaction-type** *type*] [**maxrun** *maxruntime-number*]
no event [**tag** *event-tag*] **ipsla**

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
group-name	Specifies the IP SLAs group ID.
name	Name of the IP SLAs group.
operation-id	Specifies the IP SLAs operation ID.
operation-id-value	Number in the range from 1 to 2147483647.
dest-ip-address	(Optional) Specifies the destination IP address for which the IP SLAs events are monitored.
ip-address	(Optional) Specifies the IP address of the destination port.
reaction-type	(Optional) Specifies the reaction to be taken for the specified IP SLAs operation.
type	(Optional) Type of IP SLAs reaction. One of the following keywords can be specified: • **connectionLoss** --Specifies that a reaction should occur if there is a one-way connection loss for the monitored operation. • **icpif** --Specifies that a reaction should occur if the one-way Calculated Planning Impairment Factor (ICPIF) value violates the upper threshold or lower threshold. • **jitterAvg** --Specifies that a reaction should occur if the average round-trip jitter value violates the upper threshold or lower threshold. • **jitterDSAvg** --Specifies that a reaction should occur if the average one-way destination-to-source jitter value violates the upper threshold or lower threshold. • **jitterSDAvg** --Specifies that a reaction should occur if the average one-way source-to-destination jitter value violates the upper threshold or lower threshold. • **maxOfNegativeDS** --Specifies that a reaction should occur if the one-way maximum negative jitter destination-to-source threshold is violated. • **maxOfNegativeSD** --Specifies that a reaction should occur if the one-way maximum negative jitter source-to-destination threshold is violated.

	• **maxOfPositiveDS** --Specifies that a reaction should occur if the one-way maximum positive jitter destination-to-source threshold is violated. • **maxOfPositiveSD** --Specifies that a reaction should occur if the one-way maximum positive jitter source-to-destination threshold is violated. • **mos** --Specifies that a reaction should occur if the one-way Mean Opinion Score (MOS) value violates the upper threshold or lower threshold. • **packetLateArrival** --Specifies that a reaction should occur if the one-way number of late packets violates the upper threshold or lower threshold. • **packetLossDS** --Specifies that a reaction should occur if the one-way destination-to-source packet loss value violates the upper threshold or lower threshold. • **packetLossSD** --Specifies that a reaction should occur if the one-way source-to-destination packet loss value violates the upper threshold or lower threshold. • **packetMIA** --Specifies that a reaction should occur if the one-way number of missing packets violates the upper threshold or lower threshold. • **packetOutOfSequence** --Specifies that a reaction should occur if the one-way number of packets out of sequence violates the upper threshold or lower threshold. • **rtt** --Specifies that a reaction should occur if the round-trip time violates the upper threshold or lower threshold. • **timeout** --Specifies that a reaction should occur if there is a one-way timeout for the monitored operation. • **verifyError** --Specifies that a reaction should occur if there is a one-way error verification violation.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in *ssssssss* [*.mmm*] format, where *ssssssss* must be an integer representing seconds from 0 to 31536000, and where *mmm* must be an integer representing milliseconds from 0 to 999.

Command Default

No events are published when IP SLAs operations are triggered.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

An EEM event is published when an IP SLAs reaction is triggered. Either the **group-name** or the **operation-id** must be specified. The remaining parameters are optional.

Examples

The following example shows how to publish an event when an IP SLAs operation is triggered. In this example, the group named grp1 pings the destination server 209.165.200.221 over the current interface every three seconds. If there is no response, the operation is timed out.

```
Router# configure terminal
Router(config)# event manager applet EventIPSLA
Router(config-applet)# event ipsla group-name grp1 dest-ip-address 209.165.200.221
reaction-type timeout maxrun 3
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

event manager applet

To register an applet with the Embedded Event Manager (EEM) and to enter applet configuration mode, use the **event manager applet** command in global configuration mode. To unregister the applet, use the **no** form of this command.

event manager applet *applet-name* [**authorization bypass**] [**class** *class-options*] [**trap**]
no event manager applet *applet-name* [**authorization bypass**] [**class** *class-options*] [**trap**]

Syntax Description

applet-name	Name of the applet file.
authorization	(Optional) Specifies AAA authorization type for applet.
bypass	(Optional) Specifies EEM AAA authorization type bypass.
class	(Optional) Specifies the EEM policy class.
class-options	(Optional) The EEM policy class. You can specify either one of the following: • *class-letter*-- Letter from A to Z that identifies each policy class. You can specify any one *class-letter*. • **default** --Specifies the policies registered with the default class.
trap	(Optional) Generates a Simple Network Management Protocol (SNMP) trap when the policy is triggered.

Command Default

No EEM applets are registered.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
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(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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(22)T	The **class** and **trap** keywords and the *class-options* argument were added.
15.0(1)M	The command was modified. The **authorization** and **bypass** keywords were added.

Usage Guidelines

An EEM applet is a concise method for defining event screening criteria and the actions to be taken when that event occurs.

Only one event configuration command is allowed within an applet configuration. When applet configuration submode is exited and no event command is present, a warning is displayed stating that no event is associated with this applet. If no event is specified, this applet is not considered registered and the applet is not displayed. When no action is associated with this applet, events are still triggered but no actions are performed. Multiple action applet configuration commands are allowed within an applet configuration. Use the **show event manager policy registered** command to display a list of registered applets.

Before modifying an EEM applet, use the **no** form of this command to unregister the applet because the existing applet is not replaced until you exit applet configuration mode. While you are in applet configuration mode modifying the applet, the existing applet may be executing. When you exit applet configuration mode, the old applet is unregistered and the new version is registered.

Action configuration commands are uniquely identified using the *label* argument, which can be any string value. Actions are sorted in ascending alphanumeric key sequence using the *label* argument as the sort key and are run using this sequence.

The EEM schedules and runs policies on the basis of an event specification that is contained within the policy itself. When applet configuration mode is exited, EEM examines the event and action commands that are entered and registers the applet to be run when a specified event occurs.

The EEM policies will be assigned a class when **class** *class-letter* is specified when they are registered. EEM policies registered without a class will be assigned to the **default** class. Threads that have **default** as the class will service the default class when the thread is available for work. Threads that are assigned specific class letters will service any policy with a matching class letter when the thread is available for work.

If there is no EEM execution thread available to run the policy in the specified class and a scheduler rule for the class is configured, the policy will wait until a thread of that class is available for execution. Synchronous policies that are triggered from the same input event should be scheduled in the same execution thread. Policies will be queued in a separate queue for each class using the queue_priority as the queuing order.

When a policy is triggered and if AAA is configured it will contact the AAA server for authorization. Using the **authorization bypass** keyword combination, you can skip to contact the AAA server and run the policy immediately. EEM stores AAA bypassed policy names in a list. This list is checked when policies are triggered. If a match is found, AAA authorization is bypassed.

To avoid authorization for commands configured through the EEM policy, EEM will use named method lists, which AAA provides. These named method lists can be configured to have no command authorization.

The following is a sample AAA configuration.

This configuration assumes a TACACS+ server at 192.168.10.1 port 10000. If the TACACS+ server is not enabled, configuration commands are permitted on the console; however, EEM policy and applet CLI interactions will fail.

```
enable password lab
aaa new-model
tacacs-server host 128.107.164.152 port 10000
tacacs-server key cisco
aaa authentication login consoleline none
aaa authorization exec consoleline none
aaa authorization commands 1 consoleline none
aaa authorization commands 15 consoleline none
line con 0
 exec-timeout 0 0
 login authentication consoleline
aaa authentication login default group tacacs+ enable
aaa authorization exec default group tacacs+
aaa authorization commands 1 default group tacacs+
aaa authorization commands 15 default group tacacs+
```

The **authorization**, **class** and **trap** keywords can be used in any combination.

Examples

The following example shows an EEM applet called IPSLAping1 being registered to run when there is an exact match on the value of a specified SNMP object ID that represents a successful IP SLA ICMP echo operation (this is equivalent to a **ping** command). Four actions are triggered when the echo operation fails, and event monitoring is disabled until after the second failure. A message that the ICMP echo operation to a server failed is sent to syslog, an SNMP trap is generated, EEM publishes an application-specific event, and a counter called IPSLA1F is incremented by a value of one.

```
Router(config)# event manager applet IPSLAping1
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.42.1.2.9.1.6.4 get-type exact
entry-op eq entry-val 1 exit-op eq exit-val 2 poll-interval 5
Router(config-applet)# action 1.0 syslog priority critical msg "Server IP echo failed:
OID=$_snmp_oid_val"
Router(config-applet)# action 1.1 snmp-trap strdata "EEM detected server reachability
failure to 10.1.88.9"
Router(config-applet)# action 1.2 publish-event sub-system 88000101 type 1 arg1 10.1.88.9
arg2 IPSLAEcho arg3 fail
Router(config-applet)# action 1.3 counter name _IPSLA1F value 1 op inc
```

The following example shows how to register an applet with the name one and class A and enter applet configuration mode where the timer event detector is set to trigger an event every 10 seconds. When the event is triggered, the **action syslog** command writes the message “hello world” to syslog.

```
Router(config)# event manager applet one class A
Router(config-applet)# event timer watchdog time 10
Router(config-applet)# action syslog syslog msg "hello world"
Router(config-applet)# exit
```

The following example shows how to bypass the AAA authorization when registering an applet with the name one and class A.

```
Router(config)# event manager applet one class A authorization bypass
Router(config-applet)#
```

Related Commands

Command	Description
show event manager policy registered	Displays registered EEM policies.

event manager detector routing

To set the delay time for the routing event detector to start monitoring events, use the **event manager detector routing** command in global configuration mode. To disable the delay time, use the **no** form of this command.

event manager detector routing bootup-delay *delay-time*
no event manager detector routing

Syntax Description

bootup-delay	Specifies the time delay to turn on monitoring after bootup.
delay-time	Number that represents seconds and optional milliseconds in the format sssssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.

Command Default

Routing event detector commands are not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

To configure the delay time to turn on the routing update after bootup, use the **event manager detector routing**command. If configured, the routing event detector will only start monitoring events after the bootup delay time. After the bootup delay time has been reached, the routing updates will be turned on, and the policies start will triggering.

Examples

The following example shows how to configure the delay time for the routing update to be turned on:

```
Router(config)# event manager detector routing bootup-delay 800
```

Related Commands

Command	Description
event manager detector rpc	Configures the router to accept EEM applet using RPC event detector commands.

event manager detector rpc

To configure the router to accept Embedded Event Manger (EEM) applet using remote procedure call (RPC) event detector commands, use the **event manager detector rpc**command in global configuration mode. To disable the EEM applet using the RPC event detector commands, use the **no** form of this command.

event manager detector rpc ssh [{**acl**[{[*access-list*]}] | **max-sessions** *max-sessions* | **locktime** *seconds*}]
no event manager detector rpc [{**ssh acl** [*access-list*] | **max-sessions** *max-sessions* | **locktime** *seconds*}]

Syntax Description

ssh	Specifies SSH to establish an RPC session.
acl	(Optional) Specifies an access list for this session.
access-list	(Optional) Specifies the access list for this session.
max-sessions	(Optional) Specifies the maximum number of concurrent RPC sessions.
max-sessions	(Optional) The valid range is from 4 to 16 sessions.
lock-time	(Optional) Specifies the maximum time an EEM RPC configuration lock is in place without an intermediate operation.
seconds	(Optional) The valid range is 1 to 300 seconds. The default value is 10 seconds.

Command Default

RPC event detector commands are not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

The **event manager detector rpc**commandprovides the ability to invoke EEM policies from outside the router over an encrypted connection using Secure Shell (SSH) for the highest level of security. The RPC event detector uses Simple Object Access Protocol (SOAP) data encoding for exchanging XML-based messages. This command can be used to run EEM policies and then receive output in a SOAP XML-formatted reply.

SSH must be configured on the router prior to **event manager detector rpc** is configured.

Examples

The following example shows how to configure a user, if one has not been configured. It also shows how to configure the router to enable SSHv2:

```
Router(config)# username johndoe privilege 15 password 0 lab
Router(config)# aaa new-model

Router(config)# crypto key generate rsa usage-keys label sshkeys modulus 768

Router(config)# ip ssh version 2
```

The following example shows how to connect to the router using SSH to make sure SSH is up and running:

```
Linux-server> ssh -2 -c aes256-cbc -m hmac-sha1-96 user@router
```

The following example shows how to configure the router to accept EEM RPC command, enable EEM RPC over SSHv2. You can also configure an access control list for this EEM RPC session.

```
Router# configure terminal
Router(config)# event manager detector rpc ssh acl 459
```

The following example shows how to configure the maximum time an EEM RPC configuration lock is in place without an intermediate operation. The value is set to 60 seconds.

```
Router(config)# event manager detector rpc lock-time 60
```

The following example show how to configure the maximum number of concurrent RPC sessions to 5:

```
Router(config)# event manager detector rpc max-sessions 5
```

The following example shows how to run eem_rpc via SSH.

```
Linux-server> ssh -2 -s user@172.16.0.0 eem_rpc
```

Related Commands

Command	Description
event rpc	Configures the router to accept EEM RPC.

event manager directory user

To specify a directory to use for storing user library files or user-defined Embedded Event Manager (EEM) policies, use the **event manager directory user** command in global configuration command. To disable use of a directory for storing user library files or user-defined EEM policies, use the **no** form of this command.

event manager directory user {**library** *path* | **policy** *path*}
no event manager directory user {**library** *path* | **policy** *path*}

Syntax Description

library	Specifies using the directory to store user library files.
policy	Specifies using the directory to store user-defined EEM policies.

path	Absolute pathname to the user directory on the flash device.

Command Default

No directory is specified for storing user library files or user-defined EEM policies.

Command Modes

Global configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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 user library directory is needed to store user library files associated with authoring EEM policies. If you have no plans to author EEM policies, you need not create a user library directory.

In Cisco IOS Release 12.3(14)T and later releases the software supports policy files created using the Tool Command Language (Tcl) scripting language. Tcl is provided in the Cisco IOS software image when the EEM is installed on the network device. Files with the .tcl extension can be EEM policies, Tcl library files, or a special Tcl library index file named “tclindex.” The tclindex file contains a list of user function names and the library files that contain the user functions. The EEM searches the user library directory when Tcl starts up to process the tclindex file.

To create the user library directory before identifying it to the EEM, use the **mkdir** command in privileged EXEC mode. After creating the user library directory, you can use the **copy** command to copy .tcl library files into the user library directory.

The user policy directory is needed to store user-defined policy files. If you have no plans to author EEM policies, you need not create a user policy directory. The EEM searches the user policy directory when you enter the **event manager policy** *policy-filename* **type user** command.

To create the user policy directory before identifying it to the EEM, use the **mkdir** command in privileged EXEC mode. After creating the user policy directory, you can use the **copy** command to copy policy files into the user policy directory.

Examples

The following example shows how to specify disk0:/user_library as the directory to use for storing user library files:

```
Router(config)# event manager directory user library disk0:/user_library
```

Related Commands

Command	Description
copy	Copies any file from a source to a destination.
event manager policy	Registers an EEM policy with the EEM.
mkdir	Creates a new directory in a Class C flash file system.

event manager directory user repository

To specify a default location to copy Embedded Event Manager (EEM) policy updates, use the event manager directory user repository command in global configuration mode. To disable this function, use the **no** form of this command.

event manager directory user repository *url-location*
no event manager directory user repository

Syntax Description

url-location	Location from which EEM policies will be copied.

Command Default

No directory is specified for locating EEM policies.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Examples

The following example shows how to specify tftp://10.2.2.2/user2/mktg/eem_scripts as the default location to receive EEM policies:

```
event manager directory user repository tftp://10.2.2.2/user2/mktg/eem_scripts
```

Related Commands

Command	Description
show event manager directory user repository	Displays the default directory specified to locate user EEM policy files.

event manager environment

To set an Embedded Event Manager (EEM) environment variable, use the **event manager environment** command in global configuration mode. To disable an EEM environment variable, use the **no** form of this command.

event manager environment *variable-name string*

no event manager environment *variable-name*

Syntax Description

variable-name	Name assigned to the EEM environment variable.
string	Series of characters, including embedded spaces, to be placed in the environment variable *variable-name*.

Command Default

No EEM environment variables are set.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)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(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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

By convention, the names of all environment variables defined by Cisco begin with an underscore character to set them apart: for example, _show_cmd.

To support embedded white spaces in the *string* argument, this command interprets everything after the *variable-name* argument to the end of the line to be part of the *string* argument.

To display the name and value of all EEM environment variables after you have configured them, use the **show event manager environment** command.

Examples

The following example of the **event manager environment** command defines a set of EEM environment variables:

```
Router(config)# event manager environment _cron_entry 0-59/2 0-23/1 * * 0-7
Router(config)# event manager environment _show_cmd show version
```

Related Commands

Command	Description
show event manager environment	Displays the name and value of all EEM environment variables.

event manager history size

To change the size of Embedded Event Manager (EEM) history tables, use the **event manager history size** command in global configuration mode. To restore the default history table size, use the **no** form of this command.

event manager history size {**events** | **traps**} [*size*]
no event manager history size {**events** | **traps**}

Syntax Description

events	Changes the size of the EEM event history table.
traps	Changes the size of the EEM Simple Network Management Protocol (SNMP) trap history table.
size	(Optional) Integer in the range from 1 to 50 that specifies the number of history table entries. Default is 50.

Command Default

The size of the history table is 50 entries.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)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(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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 example of the **event manager history size** command changes the size of the SNMP trap history table to 30 entries:

```
Router(config)# event manager history size traps 30
```

Related Commands

Command	Description
show event manager history events	Displays the EEM events that have been triggered.
show event manager history traps	Displays the EEM SNMP traps that have been sent.

event manager policy

To register an Embedded Event Manager (EEM) policy with the EEM, use the **event manager policy** command in global configuration mode. To unregister the EEM policy, use the **no** form of this command.

event manager policy *policy-filename* [**authorization bypass**] [**class** *class-options*] [**type** {**system** | **user**}] [**trap**]
no event manager policy *policy-filename* [**authorization bypass**] [**class** *class-options*] [**type** {**system** | **user**}] [**trap**]

Syntax Description

policy-filename	Name of the policy file.
authorization	(Optional) Specifies AAA authorization type for policy.
bypass	(Optional) Specifies EEM AAA authorization type bypass.
class	(Optional) Specifies the EEM policy class.
class-options	(Optional) The EEM policy class. You can specify either of the following: • *class-letter*-- Letter from A to Z that identifies each policy class. You can specify any one *class-letter*. • **default** --Specifies the policies registered with the default class.
type	(Optional) Specifies the type of EEM policy to be registered.
system	(Optional) Registers a Cisco-defined system policy.
user	(Optional) Registers a user-defined policy.
trap	(Optional) Generates a Simple Network Management Protocol (SNMP) trap when the policy is triggered.

Command Default

No EEM policies are registered.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.3(14)T	This command was modified. The **user** keyword was added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.

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.
12.4(22)T	This command was modified. The **class** keyword and the *class-options* argument were added.
15.0(1)M	This command was modified. The **authorization** and **bypass** keywords were added.

Usage Guidelines

The EEM schedules and runs policies on the basis of an event specification that is contained within the policy itself. When the **event manager policy** command is invoked, the EEM examines the policy and registers it to be run when the specified event occurs.

If you enter the **event manager policy** command without specifying the optional **type** keyword, the EEM first tries to locate the specified policy file in the system policy directory. If the EEM finds the file in the system policy directory, it registers the policy as a system policy. If the EEM does not find the specified policy file in the system policy directory, it looks in the user policy directory. If the EEM locates the specified file in the user policy directory, it registers the policy file as a user policy. If the EEM finds policy files with the same name in both the system policy directory and the user policy directory, the policy file in the system policy directory takes precedence and is registered as a system policy.

The EEM policies will be assigned a class when **class** *class-letter* is specified when they are registered. EEM policies registered without a class will be assigned to the **default** class. Threads that have **default** as the class will service the default class when the thread is available for work. Threads that are assigned specific class letters will service any policy with a matching class letter when the thread is available for work.

If there is no EEM execution thread available to run the policy in the specified class and a scheduler rule for the class is configured, the policy will wait until a thread of that class is available for execution. Synchronous policies that are triggered from the same input event should be scheduled in the same execution thread. Policies will be queued in a separate queue for each class using the queue_priority as the queuing order.

When a policy is triggered and if AAA is configured, it will contact the AAA server for authorization. Using the **authorization bypass** keyword combination, you can skip to contact the AAA server and run the policy immediately. EEM stores AAA-bypassed policy names in a list. This list is checked when policies are triggered. If a match is found, AAA authorization is bypassed.

To avoid authorization for commands configured through the EEM policy, EEM will use named method lists, which AAA provides. These named method lists can be configured to have no command authorization.

The following is a sample AAA configuration. This configuration assumes a TACACS+ server at 192.0.2.1 port 10000. If the TACACS+ server is not enabled, configuration commands are permitted on the console; however, EEM policy and applet CLI interactions will fail.

```
enable password lab
aaa new-model
tacacs-server host 192.0.2.1 port 10000
tacacs-server key cisco
aaa authentication login consoleline none
aaa authorization exec consoleline none
aaa authorization commands 1 consoleline none
aaa authorization commands 15 consoleline none
line con 0
 exec-timeout 0 0
 login authentication consoleline
aaa authentication login default group tacacs+ enable
aaa authorization exec default group tacacs+
aaa authorization commands 1 default group tacacs+
aaa authorization commands 15 default group tacacs+
```

The **authorization**, **class**, and **type** keywords can be used in any combination.

An error message is displayed when you try to register a ".tbc" policy that does not contain any precompiled Tool Command Language (Tcl) byte code. See the "Examples" section.

Examples

The following example shows how to use the **event manager policy** command to register a system-defined policy named tm_cli_cmd.tcl located in the system policy directory:

```
Router(config)# event manager policy tm_cli_cmd.tcl type system
```

The following example shows how to use the **event manager policy** command to register a user-defined policy named cron.tcl located in the user policy directory:

```
Router(config)# event manager policy cron.tcl type user
```

The following example shows how to use the **event manager policy** command to register a Tcl script named syslog.tcl with a class of default:

```
Router(config)# event manager policy syslog.tcl class default
```

The following example shows how to use the **event manager policy** command to register a Tcl script named syslog.tcl with a class of default and bypass the AAA authorization:

```
Router(config)# event manager policy syslog.tcl class default authorization bypass
```

The following error message is displayed when you try to register a ".tbc" policy that does not contain any precompiled TCL byte code:

```
Router(config)# event manager policy tcltotbc.tbc
EEM Register event failed: Error .tbc file does not contain compiled Tcl byte code. Error
unable to parse EEM policy for registration commands.
EEM configuration: failed to retrieve intermediate registration result for policy tcltotbc.tbc
```

Related Commands

Command	Description
show event manager policy registered	Displays registered EEM policies.

event manager run

To manually run a registered Embedded Event Manager (EEM) policy, use the **event manager run** command in privileged EXEC mode.

event manager run *policy-filename* [[*parameter1*] [*parameter2*]... [*parameter15*]]

Syntax Description

policy-filename	Name of the policy file.
parameter	(Optional) Parameter to pass to the script. A maximum of 15 parameters can be specified. The parameters must be alphanumeric strings. Do not include quotation marks, embedded spaces, and special characters.

Command Default

No registered EEM policies are run.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2SX	This command was 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	The parameter argument was added. Up to 15 parameter values can be specified, and arguments can be specified in the registry call.

Usage Guidelines

This command also enables you to use the parameters in the event policy and to specify the arguments in the registry call.

EEM usually schedules and runs policies on the basis of an event specification that is contained within the policy itself. The **event manager run** command allows policies to be run manually. The **event none** command must first be configured to run the policy manually. The None Event Detector includes arguments when it publishes the none event. This command does not have a **no** form.

Examples

The following example shows how to manually run an EEM policy named policy-manual.tcl:

```
Router# event manager run policy-manual.tcl
```

Each parameter consists of the total number of built-ins ($_none_argc), followed by the list of built-ins ($_none_arg1, $_none_arg2, and $_none_arg3). The following examples show applets and Tool Tcl scripts.

Applet Example

```
event manager applet none_parameter_test
 event none
 action 1 syslog msg "Number of Arguments is $_none_argc"
 action 2 syslog msg "Argument 1 is $_none_arg1"
 action 3 syslog msg "Argument 2 is $_none_arg2"
 action 4 syslog msg "Argument 3 is $_none_arg3"
end
Router# event manager run none_parameter_test 11 22 33
01:26:03: %HA_EM-6-LOG: none_parameter_test: Number of Arguments is 3
01:26:03: %HA_EM-6-LOG: none_parameter_test: Argument 1 is 11
```

```
01:26:03: %HA_EM-6-LOG: none_parameter_test: Argument 2 is 22
01:26:03: %HA_EM-6-LOG: none_parameter_test: Argument 3 is 33
```

For policies, **event_reqinfo**returns the optional parameters in a string, which are then handled by the policy.

Tcl Script Example

```
none_paramter_test.tcl
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
# query the event info
array set arr_einfo [event_reqinfo]
if {$_cerrno != 0} {
    set result [format "component=%s; subsys err=%s; posix err=%s;\n%s" \
        $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
action_syslog priority info msg "Number of Arguments is $arr_einfo(argc)"
if {$_cerrno != 0} {
    set result [format \
            "component=%s; subsys err=%s; posix err=%s;\n%s" \
            $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
action_syslog priority info msg "Argument 1 is $arr_einfo(arg1)"
if {$_cerrno != 0} {
    set result [format \
            "component=%s; subsys err=%s; posix err=%s;\n%s" \
            $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
action_syslog priority info msg "Argument 2 is $arr_einfo(arg2)"
if {$_cerrno != 0} {
    set result [format \
            "component=%s; subsys err=%s; posix err=%s;\n%s" \
            $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
action_syslog priority info msg "Argument 3 is $arr_einfo(arg3)"
if {$_cerrno != 0} {
    set result [format \
            "component=%s; subsys err=%s; posix err=%s;\n%s" \
            $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
    error $result
}
jubjub#event manager run none_parameter_test.tcl 1 2 3
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Number of Arguments is
 3
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Argument 1 is 1
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Argument 2 is 2
01:26:03: %HA_EM-6-LOG: tmpsys:/eem_policy/none_parameter_test.tcl: Argument 3 is 3
```

Related Commands

Command	Description
event manager applet	Registers an EEM applet with the EEM and enters applet configuration mode.

Command	Description
event manager policy	Registers an EEM policy with the EEM.
event none	Specifies that an EEM policy is to be registered with the EEM and can be run manually.
show event manager policy registered	Displays EEM policies that are already registered.

event manager scheduler

To schedule Embedded Event Manager (EEM) policies and set the policy scheduling options, use the **event manager scheduler**command in global configuration mode. To remove the scheduling of the EEM policies, use the **no** form of this command.

event manager scheduler {**applet** | **axp** | **call-home**} **thread class** *class-options* **number** *thread-number*
no event manager scheduler {**applet** | **axp** | **call-home**} **thread class** *class-options* **number** *thread-number*

Syntax Description

applet	Specifies the EEM applet policy.
axp	Specifies the application extension platform (AXP) policy.
call-home	Specifies Call Home policy.
thread	Specifies the thread for the class.
class	Specifies the EEM policy class.
class-options	The EEM policy class. You can specify either one or all of the following: • *class-letter*-- Letter from A to Z that identifies each policy class. You can specify multiple instances of *class-letter*. • **default** --Specifies the policies registered with the default class. • **range** *class-letter-range*-- Specifies the EEM policy class in a range. Multiple instances of **range** *class-letter-range* can be specified. The letters used in *class-letter-range* must be in uppercase.
number	Specifies the number of concurrent execution threads for the specified class.
thread-number	Number in the range 1 to 65535.

Command Default

Policy scheduling is active.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(22)T	This command was introduced.

Release	Modification
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

The EEM policies will be assigned a class when **class** *class-letter* is specified by the **event manager applet** or **event manager policy** commands when they are registered. EEM policies registered without a class will be assigned to the **default** class. Threads that have **default** as the class, will service the default class when the thread is available for work. Threads that are assigned specific class letters will service any policy with a matching class letter when the thread is available for work.

If there is no EEM execution thread available to run the policy in the specified class and a scheduler rule for the class is configured, the policy will wait until a thread of that class is available for execution. Synchronous policies that are triggered from the same input event should be scheduled in the same execution thread.

You should specify any one of the options *class-letter*, **default**, and **range** *class-letter-range*. You can specify all these options in the same CLI statement.

To schedule EEM policies and set the script scheduling options, use the **event manager scheduler script** command in global configuration mode. To remove the EEM script scheduling options and restore the default value, use the **no** form of this command.

Examples

The following example shows how to create two EEM execution threads to run applets of the default class.

```
Router(config)# event manager scheduler applet thread class default number 2
```

The following example shows how to create one EEM execution thread to run Tcl scripts of class A, B, D and E.

```
Router(config)# event manager scheduler script thread class A B range D-E number 1
```

Related Commands

Command	Description
event manager applet	Registers an EEM applet with the EEM and enters applet configuration mode.
event manager policy	Registers an EEM policy with the EEM.
event manager scheduler hold	Holds the EEM policy scheduling execution.
event manager scheduler script	Sets the options for the EEM script scheduling.
debug event manager scheduler suspend	Suspends the EEM policy scheduling execution.

event manager scheduler clear

To clear Embedded Event Manager (EEM) policies that are executing or pending execution, use the **event manager scheduler clear** command in privileged EXEC mode.

event manager scheduler clear {**all** | **policy** *job-id* | **queue-type** {**applet** | **call-home** | **axp** | **script**} [**class** *class-options*]} [**processor** {**rp_primary** | **rp_standby**}]

Syntax Description

all	Clears all policies that are currently executing or in the pending execution queue.
policy	Clears the EEM policy specified by the Job ID.
job-id	Number in the range from 1 to 4294967295 that identifies each policy in the queue.
queue-type	Clears the queue type of the EEM policy.
applet	Specifies the EEM queue type applet.
call-home	Specifies the EEM queue type Call Home Policies.
axp	Specifies the EEM queue type application extension platform.
script	Specifies the EEM execution thread to run the Tcl scripts.
class	Clears the EEM policies of a specified class.
class-options	Specifies the EEM policy class. You can specify either one or all of the following: • *class-letter* --Letter from A to Z that identifies each policy class. You can specify multiple instances of *class-letter*. • **default** --The default class. EEM policies registered without a class are assigned to the default class. • **range** *class-range* --Specifies the EEM policy class in a range. You can specify any range from A to Z. You can specify multiple instances of **range** *class-range*.
processor	(Optional) Specifies the processor to execute the command.
rp_primary	(Optional) Indicates the default RP. The policy is run on the primary RP when an event correlation causes the policy to be scheduled.
rp_standby	(Optional) Indicates the standby RP. The policy is run on the standby RP when an event correlation causes the policy to be scheduled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.4(22)T	The **queue-type** and **processor**keywords and *class-letter* and *class-range* arguments were added.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **show event manager policy pending** command to display the policies pending in the server execution queue.

Use the **show event manager policy active** command to display the policies that are running.

Use the **event manager scheduler clear** command to clear a policy or a policy queue in the server.

For the **class** keyword, you should specify at least one of the options, *class-letter*, **default**, or **range** *class-range*. You can specify all these options in the same CLI statement.

Examples

The following example shows how to clear EEM policies that are pending execution. The **show** commands display sample output before and after the policy is cleared.

```
Router# show event manager policy pending
no. job id status time of event             event type     name
1   1      pend   Thu Sep 7  02:54:04 2006  syslog         applet: one
2   2      pend   Thu Sep 7  02:54:04 2006  syslog         applet: two
3   3      pend   Thu Sep 7  02:54:04 2006  syslog         applet: three

Router# event manager scheduler clear policy 2
Router# show event manager policy pending

no. job id status time of event             event type     name
1   1      pend   Thu Sep 7  02:54:04 2006  syslog         applet: one
3   3      pend   Thu Sep 7  02:54:04 2006  syslog         applet: three
```

Related Commands

Command	Description
event manager policy	Registers an EEM policy with the EEM.
show event manager policy pending	Displays EEM policies that are pending execution.

event manager scheduler hold

To hold a scheduled Embedded Event Manager (EEM) policy event or event queue in the EEM scheduler, use the **event manager scheduler hold** command in privileged EXEC mode. To resume the policy event or event queue use the **event manager scheduler release** command.

event manager scheduler hold {**all** | **policy** *job-id* | **queue-type** {**applet** | **call-home** | **axp** | **script**} [**class** *class-options*]} [**processor** {**rp_primary** | **rp_standby**}]

Syntax Description

all	Holds all the EEM policy event or event queue in the EEM scheduler.
policy	Holds the EEM policy event or event queue in the EEM scheduler specified by the Job ID.
job-id	Number in the range from 1 to 4294967295 that identifies each policy in the queue.
queue-type	Holds the EEM policy event or event queue based on the EEM queue type.
applet	Specifies the EEM queue type applet.
call-home	Specifies the EEM queue type Call Home Policies.
axp	Specifies the EEM queue type application extension platform.
script	Specifies the EEM execution thread to run the Tcl scripts.
class	Specifies the class of the EEM policies.

class-options	Specifies the EEM policy class. You can specify either one or all of the following: • *class-letter* --Letter from A to Z that identifies each policy class. You can specify multiple instances of *class-letter*. • **default** --The default class. EEM policies registered without a class are assigned to the default class. • **range** *class-range* --Specifies the EEM policy class in a range. You can specify any range from A to Z. You can specify multiple instances of **range** *class-range*.
processor	(Optional) Specifies the processor to execute the command.
rp_primary	(Optional) Indicates the default RP. The policy is run on the primary RP when an event correlation causes the policy to be scheduled.
rp_standby	(Optional) Indicates the standby RP. The policy is run on the standby RP when an event correlation causes the policy to be scheduled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **show event manager policy pending** command to display the policies pending in the server execution queue.

Use the **event manager scheduler hold** command to hold a policy or a policy queue in the server.

For the **class** keyword, you should specify at least one of the options, *class-letter*, **default**, or **range** *class-range*. You can specify all these options in the same CLI statement.

Examples

The following example shows how to hold a scheduled policy event in the EEM scheduler. The **show** commands display sample output before and after the policy event is held.

```
Router# show event manager policy pending
no. job id status time of event             event type     name
1   1      pend   Thu Sep 7  02:54:04 2006  syslog         applet: one
2   2      pend   Thu Sep 7  02:54:04 2006  syslog         applet: two
3   3      pend   Thu Sep 7  02:54:04 2006  syslog         applet: three
Router# event manager scheduler hold policy 2
Router# show event manager policy pending

no. job id status time of event             event type     name
1   1      pend   Thu Sep 7  02:54:04 2006  syslog         applet: one
2  2      held   Thu Sep 7  02:54:04 2006  syslog         applet: two
3   3      pend   Thu Sep 7  02:54:04 2006  syslog         applet: three
```

Related Commands

Command	Description
event manager policy	Registers an EEM policy with the EEM.
event manager scheduler release	Resumes the policy event or event queue.
show event manager policy pending	Displays EEM policies that are pending execution.

event manager scheduler modify

To modify the scheduling parameters of the Embedded Event Manager (EEM) policies, use the **event manager scheduler modify** command in privileged EXEC mode.

event manager scheduler modify {**all** | **policy** *job-id* | **queue-type** {**applet** | **call-home** | **axp** | **script**}} {**class** *class-options* [**queue-priority** {**high** | **last** | **low** | **normal**}] | **queue-priority** {**high** | **last** | **low** | **normal**} [**class** *class-options*]} [**processor** {**rp_primary** | **rp_standby**}]

Syntax Description

all	Changes all policies that are currently executing or in the pending execution queue.
policy	Changes the EEM policy specified by the Job ID.
job-id	Number in the range from 1 to 4294967295 that identifies each policy in the queue.
queue-type	Changes the queue type of the EEM policy.
applet	Specifies the EEM queue type applet.
call-home	Specifies the EEM queue type Call Home Policies.
axp	Specifies the EEM queue type application extension platform.
script	Specifies the EEM execution thread to run the Tcl scripts.
class	Changes the class of the EEM policies.
class-options	Specifies the EEM policy class. You can specify either one or all of the following: • *class-letter* --Letter from A to Z that identifies each policy class. You can specify multiple instances of *class-letter*. • **default** --The default class. EEM policies registered without a class are assigned to the default class.
queue-priority	(Optional) Changes the priority of the queuing order of the EEM policies.
high	(Optional) Specifies the queue priority as high.
last	(Optional) Specifies the queue priority as last.
low	(Optional) Specifies the queue priority as low.
normal	(Optional) Specifies the queue priority as normal.
processor	(Optional) Specifies the processor to execute the command.

rp_primary	(Optional) Indicates the default RP. The policy is run on the primary RP when an event correlation causes the policy to be scheduled.
rp_standby	(Optional) Indicates the standby RP. The policy is run on the standby RP when an event correlation causes the policy to be scheduled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **show event manager policy pending** command to display the policies pending in the server execution queue.

Use the **event manager scheduler modify** command to modify the scheduling parameters of a policy.

For the **class** keyword, you should specify at least one of the options, *class-letter* or **default**. You can specify both the options in the same CLI statement.

Examples

The following example shows how to modify the scheduling parameters of the EEM policies. The **show** commands display sample output before and after the scheduling parameters are modified.

```
Router# show event manager policy pending
no. class   status time of event             event type     name
1   default pend   Thu Sep 7  02:54:04 2006  syslog         applet: one
2   default pend   Thu Sep 7  02:54:04 2006  syslog         applet: two
3   B       pend   Thu Sep 7  02:54:04 2006  syslog         applet: three

Router# event manager scheduler modify all class A
Router# show event manager policy pending

no. class status time of event             event type     name
1   A     pend   Thu Sep 7  02:54:04 2006  syslog         applet: one
2   A     pend   Thu Sep 7  02:54:04 2006  syslog         applet: two
3   A     pend   Thu Sep 7  02:54:04 2006  syslog         applet: three
```

Related Commands

Command	Description
event manager policy	Registers an EEM policy with the EEM.
show event manager policy pending	Displays EEM policies that are pending execution.

event manager scheduler release

To resume execution of the specified Embedded Event Manager (EEM) policies, use the **event manager scheduler release**command in privileged EXEC mode.

event manager scheduler release {**all** | **policy** *policy-id* | **queue-type** {**applet** | **call-home** | **axp** | **script**} [**class** *class-options*]} [**processor** {**rp_primary** | **rp_standby**}]

Syntax Description

all	Resumes the execution of all EEM policies.
policy	Resumes the EEM policy specified by the policy ID.
policy-id	Number in the range from 1 to 4294967295 that identifies each policy in the queue.
queue-type	Resumes the execution of policies based on the EEM queue type.
applet	Specifies the EEM applet policy.
call-home	Specifies the Call Home policy.
axp	Specifies the application extension platform (AXP) policy.
script	Specifies the EEM script policy.
class	Specifies the EEM policy class.
class-options	The EEM policy class. You can specify either one or all of the following: • *class-letter*-- Letter from A to Z that identifies each policy class. You can specify multiple instances of *class-letter*. • **default** --Specifies the policies registered with the default class. • **range** *class-letter-range*-- Specifies the EEM policy class in a range. Multiple instances of **range** *class-letter-range* can be specified. The letters used in *class-letter-range* must be in uppercase.
processor	Specifies the processor to execute the command. The default value is the primary RP.
rp_primary	Indicates the primary RP.
rp_standby	Indicates the standby RP.

Command Default

Disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)T	This command was introduced.

Usage Guidelines

To release the EEM policies held using the **event manager scheduler hold** command, use the **event manager scheduler release** command.

You should specify any one of the options *class-letter*, **default**, and **range** *class-letter-range*. You can specify all these options in the same CLI statement.

Examples

The following example shows how to resume the execution of all the EEM policies:

```
Router# event manager scheduler release all
```

The following example shows how to resumes the execution for policies of class A to E:

Router# **event manager scheduler release queue-type script class range A-E**

Related Commands

Command	Description
event manager scheduler hold	Holds the EEM policy scheduling execution.

event manager scheduler script

To schedule Embedded Event Manager (EEM) policies and set the script scheduling options, use the **event manager scheduler script** command in global configuration mode. To remove the EEM script scheduling options and restore the default value, use the **no** form of this command.

event manager scheduler script thread class *class-options* **number** *thread-number*
no event manager scheduler script thread class *class-options* **number** *thread-number*

Syntax Description

thread	Specifies the thread for the class.
class	Specifies the EEM policy class.
class-options	The EEM policy class. You can specify either one or all of the following: • *class-letter*-- Letter from A to Z that identifies each policy class. You can specify multiple instances of *class-letter*. • **default** --Specifies the policies registered with the default class. • **range** *class-letter-range*-- Specifies the EEM policy class in a range. Multiple instances of **range** *class-letter-range* can be specified. The letters used in *class-letter-range* must be in uppercase.
number	Specifies the number of concurrent execution threads for the specified class.
thread-number	Number in the range 1 to 65535.

Command Default

Only one EEM policy can be run at a time.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.

Release	Modification
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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(22)T	The **range** and **number**keywords and *class-options* argument were added.

Usage Guidelines

Use the **event manager scheduler script** command if you want to run more than one EEM policy concurrently.

You should specify any one of the options *class-letter*, **default**, and **range** *class-letter-range*. You can specify all these options in the same CLI statement.

To schedule EEM policies and set the policy scheduling options, use the **event manager scheduler**command in global configuration mode. To remove the scheduling of the EEM policies, use the **no** form of this command.

Examples

The following example shows how to specify two script execution threads to run concurrently:

```
Router(config)# event manager scheduler script thread class default number 2
```

event manager scheduler suspend

To immediately suspend Embedded Event Manager (EEM) policy scheduling execution, use the **event manager scheduler suspend** command in global configuration mode. To resume EEM policy scheduling, use the **no** form of this command.

event manager scheduler suspend
no event manager scheduler suspend

Syntax Description

This command has no arguments or keywords.

Command Default

Policy scheduling is active.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)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.

Release	Modification
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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 **event manager scheduler suspend**command to suspend all policy scheduling requests and do no scheduling until you enter the **no** form of the command. The **no** form of the command resumes policy scheduling and executes any pending policies.

You might want to suspend policy execution immediately instead of unregistering policies one by one for the following reasons:

- For security--if you think the security of your system has been compromised.
- For performance--if you want to suspend policy execution temporarily to make more CPU cycles available for other functions.

Examples

The following example of the **event manager scheduler suspend**command disables policy scheduling:

```
Router(config)# event manager scheduler suspend
May 19 14:31:22.439: fm_server[12330]: %HA_EM-6-FMS_POLICY_EXEC: fh_io_msg: Policy execution
 has been suspended
```

The following example of the **event manager scheduler suspend**command enables policy scheduling:

```
Router(config)# no event manager scheduler suspend
May 19 14:31:40.449: fm_server[12330]: %HA_EM-6-FMS_POLICY_EXEC: fh_io_msg: Policy execution
 has been resumed
```

Related Commands

Command	Description
event manager policy	Registers an EEM policy with the EEM.

event manager session cli username

To associate a username with Embedded Event Manager (EEM) policies that use the command-line interface (CLI) library, use the **event manager session cli usernameprivilege<0-15>** command in global configuration mode. To remove the username association with EEM policies that use the CLI library, use the **no** form of this command.

event manager session cli username *username***privilege<0-15>**
no event manager session cli username *username***privilege<0-15>**

Syntax Description

username	Username assigned to EEM CLI sessions that are initiated by EEM policies.

Command Default

No username is associated with EEM CLI sessions.

Command Modes

Global configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
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 **event manager session cli username** command to assign a username for EEM policy CLI sessions when TACACS+ is used for command authorization.

If you are using authentication, authorization, and accounting (AAA) security and implement authorization on a command basis, you should use the **event manager session cli username** command to set a username to be associated with a Tool Command Language (Tcl) session. The username is used when a Tcl policy executes a CLI command. TACACS+ verifies each CLI command using the username associated with the Tcl session that is running the policy. Commands from Tcl policies are not usually verified because the router must be in privileged EXEC mode to register the policy.

Examples

The following example of the **event manager session cli username** command associates the username eemuser with EEM CLI sessions initiated by EEM policies:

```
Router(config)# event manager session cli username eemuser
```

Related Commands

Command	Description
show event manager session cli username	Displays the username associated with CLI sessions initiated by EEM policies that use the EEM CLI library.

event manager update user policy

To specify an immediate Embedded Event Manager (EEM) policy update, use the event manager update user policy command in privileged EXEC mode.

event manager update user policy [{**name** **policy-filename** | **group** **group-name-exp**}] [**repository** **url-location**]

Syntax Description

name policy-filename	(Optional) Specifies the EEM policy name.
group group-name-exp	(Optional) Specifies the EEM policy group name.
repository url-location	(Optional) Specifies the EEM policy repository directory. The *url-location* argument is the location from which EEM policies will be copied. The default repository is that set in **event manager directory user repository command.**

Command Default

No EEM policies are registered.

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 Release 12.2(33)SRE.

Usage Guidelines

The event manager update user policy command is used for the following purposes:

- To specify a single policy using the name policy-filename option. An attempt will be made to copy the specified policy from the user policy repository URL to the current user policy directory. If successful, a check will be made to see if the policy is registered and, if so, it will be unregistered. Then the newly copied policy will be registered.
- To specify a regular expression pattern string to match a group of policies using the group group-name-exp option. An attempt will be made to copy all registered policies whose policy names match the specified regular expression from the user policy repository URL to the current user policy directory. If successful, they will be unregistered and the newly downloaded policies will be registered.

Note If an error occurs registering a newly downloaded policy, the policy that was previously registered will be left unregistered.

Note If the repository URL is the same as the user policy directory URL, the copy step will be skipped and the policy will be unregistered then reregistered.

All activities will be logged to the CLI EXEC session and syslog.

Examples

The following example shows policy sl_intf_down.tcl specified from the tftp://10.2.2.2/users2/mktg/eem_ repository:

```
event manager update user policy name sl_intf_down.tcl repository
tftp://10.2.2.2/users2/mktg/eem_
```

The following example shows a group of policies specified from the tftp://10.2.2.2/users2/mktg/eem_ repository:

```
event manager update user policy group “*.tcl” repository tftp://10.2.2.2/users2/mktg/eem_
```

Related Commands

Command	Description
event manager directory user repository	Specifies a default location from which to receive EEM policy updates.