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 application**command 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 sssssss[.mmm] format, where sssssss 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.
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 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 sssssss[.mmm] format, where sssssss 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 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.

Release	Modification
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 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 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 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.

Examples

```
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.

Examples

```
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.
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 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.

Command	Description
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.

Command	Description
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 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.
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.