Embedded Event Manager Command Reference, Cisco IOS XE Release 3SE (Catalyst 3850 Switches)

First Published: January 22, 2013

Last Modified: January 22, 2013

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

CONTENTS

A through action snmp Commands

A through action snmp Commands

action add

To specify the action of adding values of two variables when an an Embedded Event Manager (EEM) applet is triggered, use the **action add** command in applet configuration mode. To undo the add action, use the **no** form of this command.

action *label* **add** {*long-integer*| *variable-name*} {*long-integer*| *variable-name*}

no action *label* **add**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
add	Adds the values of two variables.
variable-name	String value to be placed as the variable name.
long-integer	Long integer value to be added to a variable.

Command Default

By default, there is no change in the value of variables configured within an EEM applet.

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

You can use this command to add the values of two variables. The result is stored in the variable named $_result. The value of the variable must be a long integer, else the action will fail.

Examples

The following example shows how to configure an EEM applet to add the values of two variables:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set $var1 10
Router(config-applet)#action 1.0 set $var2 20
Router(config-applet)#action 1.0 add $var1 $var2
Router(config-applet)#
```

Related Commands

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

action append

To specify the action of appending the given string value to the current value of a variable when an Embedded Event Manager (EEM) applet is triggered, use the **action append** command in applet configuration mode. To undo the append action, use the **no** form of this command.

action *label* **append** *variable-name* [*variable-value*]

no action *label* **add**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
append	Appends the given string value to the current value of the variable specified.
variable-name	String value to be placed as the variable name.
variable-value	(Optional) Long integer value to be appended tothe value of the variable name specified.

Command Default

By default, there is no change in the value of variables configured within an EEM applet.

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

You can use this command to append the given string value to the current value of variable. If the variable does not exist, it will be created and set to the given value.

Examples

The following example shows how to configure an EEM applet to append given string value to the current value of the variable specified:

```
Router(config)#event manager applet one
```

```
Router(config-applet)#action 1.0 set $var1 10
Router(config-applet)#action 1.0 append $var1 12
Router(config-applet)#
```

Related Commands

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

action break

To specify the action of exiting from a loop of actions when an Embedded Event Manager (EEM) applet is triggered, use the **action break** command in applet configuration mode. To disable the break action, use the **no** form of this command.

action *label* **break**

no action *label* **break**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
break	Causes an immediate exit from a loop of actions.

Command Default

By default, there is no exit from a loop of actions configured within an EEM applet.

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

You can use this command to skip all the actions down to the related end action.

Examples

The following example shows how to configure an EEM applet to break from a loop of actions:

```
Router(config)# event manager applet loop
Router(config-applet)# event none
Router(config-applet)# action 1 while 1 eq 1
Router(config-applet)# action 2 break
Router(config-applet)# action 3 end
```

Related Commands

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

action cli

To specify the action of executing a Cisco IOS command-line interface (CLI) command when an Embedded Event Manager (EEM) applet is triggered, use the **action cli**command in applet configuration mode. To remove the action of executing a CLI command, use the **no** form of this command.

action *label* **cli command** *cli-string* [**pattern** *pattern-string*]

no action *label* **cli command** *cli-string*

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
command	Specifies the message to be sent to the Cisco IOS CLI.
cli-string	CLI command to be executed. If the string contains embedded blanks, enclose it in double quotation marks.
pattern	(Optional) Specifies the regular expression response pattern for the **command** *cli-string*only when the command string solicits input.
pattern-string	(Optional) Specifies the action to be specified with the **pattern** keyword. You are required to specify a regular expression *pattern-string* that will match the next solicited prompt.

Command Default

No CLI commands are executed when an EEM applet is triggered.

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.

Release	Modification
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.
12.2(33)SXH	The **pattern** keyword was added.

Usage Guidelines

Use the **action cli** command to specify the action of executing a Cisco IOS CLI command when an EEM applet is triggered. The **pattern** keyword is optional and is used only when the command string solicits input.

There are two types of Cisco IOS CLI commands:

- Normal--Those Cisco IOS CLI commands that produce output followed by a display of the normal router prompt. The **action cli** command ends when the normal router prompt is received.
- Solicited--Those Cisco IOS CLI commands that ask one or more questions before the normal router prompt is displayed, such as “confirm,” which has to be completed with a “yes” or a “no”input.

The **action cli**command ends when the solicited prompt as specified in the optional **pattern** keyword is received. You are required to specify a regular expression pattern that will match the next solicited prompt. Specifying an incorrect pattern will cause the **action cli** command to wait forever until the applet execution times out due to the maxrun timer expiration.

The vty lines are allocated from the pool of vty lines that are configured using the **line vty** CLI configuration command. EEM will use a vty line when a vty line is not being used by EEM and there are available vty lines. EEM will also use a vty line when EEM is already using a vty line and there are three or more vty lines available. Be aware that the connection will fail when fewer than three vty lines are available, preserving the remaining vty lines for Telnet use.

The table below shows the built-in variable that is set when the **action cli** command is run.

Table 1: EEM Built-in Variables for action cli Command

Built-in Variable	Description
$_cli_result	The result of the execution of the CLI command.

Examples

The following example shows how to specify an EEM applet to run when the Cisco IOS **interface loopback** CLI command is configured three times. The applet executes the **no shutdown** command to ensure that the loopback interfaces are operational.

```
Router(config)# event manager applet cli-match
Router(config-applet)# event cli command {.*interface loopback*} sync yes occurs 3
Router(config-applet)# action 1.0 cli command "no shutdown"
```

The following example shows how to specify an EEM applet to run when the **pattern** keyword specifies the *confirm* argument for the **clear counters Ethernet0/1** command.

```
Router(config)# event manager applet cli-match
Router(config-applet)# action 1.0 cli command "enable"
Router(config-applet)# action 2.0 cli command "clear counters Ethernet0/1" pattern "confirm"
```

```
Router(config-applet)# action 3.0 cli command "y"
!
```

Related Commands

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

action comment

To specify the action of adding comments to an applet when an Embedded Event Manager (EEM) applet is triggered, use the **action comment** command in applet configuration mode. To disable the comment, use the **no** form of this command.

action *label* **comment** *string*

no action *label* **comment**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
comment	Adds comments to an applet.
string	Series of characters, including embedded spaces, to be placed as the comment.

Command Default

By default, there are no comments added to an applet.

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

You can use this command to add comments to applets. This results in a no-op when the applet is run.

Examples

The following example shows how to add comments to an applet:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 comment keyvalue
Router(config-applet)#
```

Related Commands

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

action context retrieve

To specify the action of retrieving variables identified by a given set of context name keys, when an Embedded Event Manager (EEM) applet is triggered, use the **action context retrieve**command in applet configuration mode. To undo the retrieve action, use the **no** form of this command.

action *label* **context retrieve key** *key-name* **variable** *variable-name-pattern*

no action *label* **context retrieve**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
context retrieve	Used to retrieve variables identified by the given context name keys.
key *key-name*	Provides the context name key.
variable *variable-name-pattern*	Provides description of the variable.

Command Default

By default, no variables specified by a given set of context name keys are retrieved.

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

You can use this command to retrieve the variable(s) identified by a given set of context name keys. Information that is retrieved is automatically deleted from the context database.

The information for the variable specified in the command is retrieved, only if a variable with the same name was saved in the corresponding context save call, using the **action context save**command.

Examples

The following example shows how to configure an EEM applet to retrieve variables identified by a given set of context name keys:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 context retrieve key pki-72a variable var1
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
action context save	This command is used to save information across multiple policy triggers.

action context save

To specify the action of saving information across multiple policy triggers, when an Embedded Event Manager (EEM) applet is triggered, use the **action context save**command in applet configuration mode. To remove the saved the information, use the **no** form of this command.

action *label* **context save key** *key-name* **variable** *variable-name-pattern*

no action *label* **context save**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
context save	Used to save information across multiple policy triggers.
key *key-name*	Provides the context name key.
variable *variable-name-pattern*	Provides description of the variable.

Command Default

By default, no infomration in saved across multiple policy triggers.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

You can use the **action context save**command to save information across multiple policy triggers. The command saves variables that match the given pattern with the context name key as identification. Saved information can be retrieved by a different applet using the **action context retrieve**command.

Once the saved information is retrieved, it is automatically deleted from the context database. To save the same information from the applet that retrieved it, you must run the **action context save**command on that applet again.

Examples

The following example shows how to configure an EEM applet to save information across multiple policy triggers:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 context save key pki-72a variable var1
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
action context retrieve	Retrieves variables identified by the given context name keys.

action else

To identify the beginning of an else conditional action block in an if/else conditional action block when an Embedded Event Manager (EEM) applet is triggered, use the **action else** command in applet configuration mode. To remove the else conditional action block, use the **no** form of this command.

action *label* **else**

no action *label* **else**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

If the command is not entered within applet configuration mode, the respective applet is not registered when you exit the configuration.

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

Use the **action else**command to identify the else conditional action block. If a statement is not associated with this applet, events are still triggered without any action or result. A warning message stating that no statements are associated with this applet is displayed at the exit time of the configuration.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- Event specification criteria are written in Tcl in the Tcl-based implementation.
- Event specification data are written using the CLI applet-submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet-based implementation because the Tcl-based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name* command after entering global configuration mode. In applet configuration mode the config prompt changes to (config-applet)#. The applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that causes this applet to run.
- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows how to identify the beginning of an else action block:

```
Router(config)# event manager applet action
Router(config-applet)# action label if $var eq 0
Router(config-applet)# action label2 else
Router(config-applet)# end
```

Related Commands

Command	Description
action elseif	Identifies the beginning of an elseif conditional action block when an EEM applet is triggered.
action if	Identifies the beginning of an if conditional action block when an EEM applet is triggered.

action end

To identify the end of a conditional action block in the if/else and while conditional action block when an Embedded Event Manager (EEM) applet is triggered, use the **action end** command in applet configuration mode. To remove the end conditional action block, use the **no** form of this command.

action *label* **end**

no action *label* **end**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

If the commands are not specified within applet configuration mode, the respective applet is removed when you exit the configuration.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

Use the **action end**command to identify the end of a conditional action block in the if/else and while conditional action block.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- Event specification criteria are written in Tcl in the Tcl-based implementation.
- Event specification data are written using the CLI applet submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet-based implementation because the Tcl-based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name* command after entering global configuration mode. In applet configuration mode the config prompt changes to (config-applet)#. The applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that causes this applet to run.

- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows hoe to identify the end of a conditional action block:

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1.0 set x "5"
Router(config-applet)# action 2.0 if $x lt 10
Router(config-applet)# action 3.0  puts "$x is less than 10"
Router(config-applet)# action 4.0 end
```

Related Commands

Command	Description
action else	Identifies the beginning of the else conditional action block when an EEM applet is triggered.
action if	Identifies the beginning of an if conditional action block when an EEM applet is triggered.

action foreach

To specify the iteration of an input string using the delimiter as a tokenizing pattern, use the **action foreach** command in applet configuration mode. To remove iteration of the input string, use the **no** form of this command.

action *label* **foreach** [*string-iterator*] [*string-input*] [*string-delimiter*]

no action *label* **foreach**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string-iterator	(Optional) Series of characters that acts as an iterator. If the string contains embedded blanks, enclose it in double quotation marks.
string-input	(Optional) Series of characters that acts as an input. If the string contains embedded blanks, enclose it in double quotation marks.
string-delimiter	(Optional) Series of characters that acts as a delimiter. If the string contains embedded blanks, enclose it in double quotation marks. The default delimiter is whitespace.

Command Default

If the commands are not specified within applet configuration mode, the respective applet is removed when you exit the configuration.

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

Use the **action foreach** command to iterate an input string using the delimiter as a tokenizing pattern. The delimiter is a regular expression pattern string. The token found in each iteration is assigned to the given

iterator variable. All arithmetic calculations are performed as long integers without any checks for overflow. If a statement is not associated with this applet, events are still triggered without any action or result. A warning message stating that no statements are associated with this applet is displayed at the exit time of the configuration.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- The event specification criteria are written in Tcl in the Tcl-based implementation.
- The event specification data are written using the CLI applet submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet-based implementation because the Tcl-based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name*command after entering global configuration mode. In applet configuration mode the config prompt changes to (config-applet)#. The applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that causes this applet to run.
- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows how to iterate an input string using the delimiter as a tokenizing pattern:

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1 foreach _iterator "red blue green orange"
Router(config-applet)# action 2  puts "iterator is $_iterator"
Router(config-applet)# action 3 end
Router# event manager run action
iterator is red
iterator is blue
iterator is green
iterator is orange
Router#
```

action gets

To get an input from the local tty in a synchronous applet and store the value in the given variable when an Embedded Event Manager (EEM) applet is triggered, use the **action gets** command in applet configuration mode. To cancel the process of receiving an input from the local tty, use the **no** form of this command.

action *label* **gets** *variable*

no action *label* **gets**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
variable	Variable word that stores the input value from the synchronous applet.

Command Default

If the commands are not specified within applet configuration mode, the respective applet is removed when you exit the configuration.

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

Use the **action gets** command to get an input from the local tty in a synchronous applet and store the value in the given variable. This command is not operational for asynchronous applets. The applet continues without any error but does not set the variable. All arithmetic calculations are performed as long integers without any checks for overflow. If a statement is not associated with this applet, events are still triggered without any action or result. A warning message stating that no statements are associated with this applet is displayed at the exit time of the configuration.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- Event specification criteria are written in Tcl in the Tcl-based implementation.

- Event specification data are written using the CLI applet submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet based implementation because the Tcl-based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name* command after entering the global configuration mode. In applet configuration mode the config prompt changes to (config-applet)#. The applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that causes this applet to run.
- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows how to get the input from the local tty in a synchronous applet and store the value:

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action label2 gets input
Router(config-applet)# action label3 syslog msg “Input entered was \”$input\“”
```

Related Commands

Command	Description
action puts	Prints data directly to the local tty in a synchronous applet when an EEM applet is triggered.

action if

To identify the beginning of an if conditional block when an Embedded Event Manager (EEM) applet is triggered, use the **action if** command in applet configuration mode. To remove the if conditional action block, use the **no** form of this command.

action *label* **if** [*string-op-1*] {**eq**| **gt**| **ge**| **lt**| **le**| **ne**} [*string-op-2*]

no action *label* **if**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string-op-1	(Optional) Sequence of characters that will replace the range of characters in the string.
eq	Equal To keyword used for comparing two strings.
gt	Greater Than keyword used for comparing two strings.
ge	Greater Than or Equal To keyword used for comparing two strings.
lt	Less Than keyword used for comparing two strings.
le	Less Than or Equal To keyword used for comparing two strings.
ne	Not Equal To keyword used for comparing two strings.
string-op-2	(Optional) Sequence of characters that will replace the range of characters in the string.

Command Default

If the commands are not specified within applet configuration mode, the respective applet is removed when you exit the configuration.

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

Use the **action if**command to identify the beginning of the if conditional action block. All arithmetic calculations are performed as long integers without any checks for overflow. If the *goto label* option is used, the if functionality will not identify the beginning of a action block, but will signify the applet to jump to the given label if the condition is true.

If a statement is not associated with this applet, events are still triggered without any action or result. A warning message stating that no statements are associated with this applet is displayed at the exit time of the configuration.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- Event specification criteria are written in Tcl in the Tcl-based implementation.
- Event specification data are written using the CLI applet submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet based implementation because the Tcl based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name* command after entering global configuration mode. In applet configuration mode the config prompt changes to (config-applet)#. The applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that causes this applet to run.
- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows how to identify the beginning of an if conditional block:

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1.0 set x "5"
Router(config-applet)# action 2.0 if $x lt 10
Router(config-applet)# action 3.0  puts "$x is less than 10"
Router(config-applet)# action 4.0 end
Router# event manager run action
5 is less than 10
Router#
```

Related Commands

Command	Description
action elseif	Identifies the beginning of the else conditional action block in the else/if conditional block when an EEM applet is triggered.
action ifgoto	Signifies the applet to jump to the given label if the condition is true when an EEM applet is triggered.

action ifgoto

To instruct the applet to jump to a given label if the specified condition is true when an Embedded Event Manager (EEM) applet is triggered, use the **action ifgoto** command in applet configuration mode. To cancel the process of applet jump, use the **no** form of this command.

action *label-1* **if** [*string-op-1*] {**eq**| **gt**| **ge**| **lt**| **le**| **ne**} [*string-op-2*] **goto** *label-2*

no action *label* **ifgoto**

Syntax Description

label-1	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string-op-1	(Optional) Sequence of characters that will replace the range of characters in the string.
eq	Equal To keyword used for comparing two strings.
gt	Greater Than keyword used for comparing two strings.
ge	Greater Than Or Equal To keyword used for comparing two strings.
lt	Less Than keyword used for comparing two strings.
le	Less Than Or Equal To keyword used for comparing two strings.
ne	Not Equal To keyword used for comparing two strings.
string-op-2	(Optional) Sequence of characters that will replace the range of characters in the string.
label-2	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

If the command is not specified within applet configuration mode, the respective applet is removed when you exit the configuration.

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

Use the **action ifgoto**command to signify the applet to jump to a given label if the specified condition is true. If the **goto** *label*option is used, the **action if** command will not identify the beginning of an action block. Goto actions are supported only within the if/goto format. To simulate a goto without if, use a test that is always true. All arithmetic calculations are performed as long integers without any checks for overflow. If a statement is not associated with this applet, events are still triggered without any action or result. A warning message stating that no statements are associated with this applet is displayed at the exit time of the configuration.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- Event specification criteria are written in Tcl in the Tcl-based implementation.
- Event specification data is written using the CLI applet submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet-based implementation because the Tcl-based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name* command in the global configuration mode. In applet configuration mode, the config prompt changes to (config-applet)#. Applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that cause this applet to run.
- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows how to instruct the applet to jump to a given label:

```
Router(config)# event manager applet action
Router(config-applet)# event none
Router(config-applet)# action 1 set x "5"
Router(config-applet)# action 2 if $x lt 10 goto 4
Router(config-applet)# action 3 puts "skipping this"
Router(config-applet)# action 4 puts "jumped to action 4"
Router(config-applet)# action 5 end
Router# event manager run action
jumped to action 4
```

Related Commands

Command	Description
action else	Identifies the beginning of the else conditional action block when an EEM applet is triggered.
action if	Identifies the beginning of an if conditional action block when an EEM applet is triggered.

action increment

To specify the action of incrementing the value of a variable, when an Embedded Event Manager (EEM) applet is triggered, use the **action increment** command in applet configuration mode. To remove the action from the applet, use the **no** form of this command.

action *label* **increment** *variable-name long-integer*

no action *label* **increment**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
increment	Increments the value of the variable with the long integer specified.
variable-name	String value to be placed as the variable name.
long-integer	Long integer value by which the variable is incremented.

Command Default

By default, there is no change in the value of variables configured within an EEM applet.

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

You can use this command to increment the value of a variable. The value of the variable must be a long integer, else the action will fail.

Examples

The following example shows how to configure an EEM applet to increment the value of a variable:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set varname 20
```

```
Router(config-applet)#action 1.0 increment varname 12
Router(config-applet)#
```

Related Commands

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

action info type interface-names

To obtain interface names when an Embedded Event Manager (EEM) applet is triggered, use the **action info type interface-names**command in applet configuration mode. To disable the action of obtaining interface names, use the **no** form of this command.

action *label* **info type interface-names** [**include** *string-operator*| **exclude** *string-operator*| **regexp** *regular-expression*]

no action *label* **info type**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
include	(Optional) Includes all interface names that contain the string pattern.
exclude	(Optional) Excludes all interface names that contain the string pattern.
string-operator	(Optional) String pattern for including or excluding the interface names.
regexp	(Optional) Obtains all the interfaces that match the specified regular expression.
regular-expression	(Optional) Regular expression pattern. For example, [^abc].

Command Default

All the current interface names are obtained from the database.

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

The **action info type interface-names**command obtains the current interface names and stores them as a space-separated list in the $_info_interface_names built-in variable.

Examples

The following example shows how to specify that interface names that include “eth” are obtained:

```
Router# configure terminal
Router(config)# event manager applet interface-app
Router(config-applet)# action 1.2 info type interface-names include eth
```

Related Commands

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

action info type snmp getid

To retrieve the individual variables from a Simple Network Management Protocol (SNMP) entity during the SNMP get operation, use the **action info type snmp getid**command in applet configuration mode. To disable the retrieving of individual variables from SNMP, use the **no** form of this command.

action *label* **info type snmp getid** *oid-value* [**community** *community-string*] [**ipaddr** *ip-address*]

no action *label* **info type**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
getid	Retrieves SNMP variables.
oid-value	Object ID value of the data element, in SNMP dotted notation. An object identifier is expressed as a series of integers or text strings. For example, the object name for the interfaces MIB can be expressed as 1.3.6.1.2.1.2 or iso.internet.mgmt.mib-2.interfaces. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. The following types are valid: • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE
community	(Optional) Specifies the community string to access the SNMP entity.

community-string	(Optional) SNMP community string. Community string functions like passwords to access the SNMP entity. The string can consist of 1 to 32 alphanumeric characters and can be set to any of the following types of community strings: • **ro** --Sets the read-only access to the SNMP entity. The default value for this community string is **public**. • **rw** --Sets read-write access to the SNMP entity. The default value for this community string is **private**.
ipaddr	(Optional) Specifies the IP address of the SNMP entity.
ip-address	(Optional) IP address of Network Management System (NMS) from which the objects are retrieved for SNMP get and set operations.

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

The table below shows the built-in variables in which the variables retrieved from the SNMP get operation are stored.

Table 2: EEM Built-in Variables for action info Command

Built-in Variable	Description
$_info_snmp_sysname_oid	The OID value of the sysName variable.
$_ info_snmp_sysname_value	The value string for the sysName variable.
$_info_snmp_syslocation_oid	The OID value of the sysLocation variable.
$_info_snmp_syslocation_value	The value string for the sysLocation variable.
$_info_snmp_sysdescr_oid	The OID value of the sysDescr variable.
$_info_snmp_sysdescr_value	The value string for the sysDescr variable.

Built-in Variable	Description
$_info_snmp_sysobjectid_oid	The OID value of the sysObjectID variable.
$_info_snmp_sysobjectid_value	The value string for the sysObjectID variable.
$_info_snmp_sysuptime_oid	The OID value of the sysUptime variable.
$_info_snmp_sysuptime_value	The value string for the sysUptime variable.
$_info_snmp_syscontact_oid	The OID value of the sysContact variable.
$_info_snmp_syscontact_value	The value string for the sysContact variable.

Examples

The following example shows how to retrieve the sysDescr.0 variable from an SNMP entity:

```
Router(config)# event manager applet
Router(config-applet)# action 1.3 info type snmp getid 1.3.6.1.2.1.1.1.0 community public
ipaddr 172.17.16.69
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.
snmp-server community	Sets the community access string to enable access to the SNMP entity.

action info type snmp inform

To send Simple Network Management Protocol (SNMP) inform requests when an Embedded Event Manager (EEM) applet is triggered, use the **action info type snmp inform**command in applet configuration mode. To disable the sending of SNMP inform requests, use the **no** form of this command.

action *label* **info type snmp inform trap-oid** *trap-oid-value* **trap-var** *trap-variable* **community** *community-string* **ipaddr** *ip-address*

no action *label* **info type**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
trap-oid	Specifies the object identifier of the object generating the SNMP trap.
trap-oid-value	The OID value of the object generating the SNMP trap.
trap-var	Specifies the variable associated with the instance of the object generating the trap.
trap-variable	The variable value of the object generating SNMP trap.
community	Specifies the community string to access the SNMP entity.
community-string	SNMP community string. Community string functions like passwords to access the SNMP entity. The string can consist of 1 to 32 alphanumeric characters and can be set to any of the following: • **ro** --Sets the read-only access to the SNMP entity. The default value for this community string is **public**. • **rw** --Sets read-write access to the SNMP entity. The default value for this community string is **private**.
ipaddr	Specifies the IP address of the SNMP entity.

ip-address	IP address of Network Management System (NMS) from which the objects are retrieved for SNMP get and set operations.

Command Default

No SNMP inform requests are sent by default.

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

SNMP inform requests are the SNMP notifications that alert the SNMP manager to a network condition and request for confirmation of receipt from the SNMP manager.

Examples

The following example shows how to send an SNMP inform request:

```
Router(config)# event manager applet
Router(config-applet)# action 1.4 info type snmp inform trap-oid 1.3.6.1.4.1.1.226.0.2.1
trap-var sysUpTime community public ipaddr 172.69.16.2
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.
snmp-server community	Sets the community access string to enable access to the SNMP entity.
snmp-server enable traps	Enables all SNMP notification types available on your system.

action info type snmp oid

To specify the type of Simple Network Management Protocol (SNMP) get operation and the object to retrieve during the SNMP set operation, when an Embedded Event Manager (EEM) applet is triggered, use the **action info type snmp oid**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **info type snmp oid** *oid-value* {**get-type** {**exact**| **next**} [**community** *community-string*]| **set-type** *oid-type oid-type-value* **community** *community-string*} [**ipaddr** *ip-address*]

no action *label* **info type**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
oid	Requests the value of the SNMP object as specified by the SNMP object identifier (OID).
oid-value	Object ID value of the data element, in SNMP dotted notation. An object identifier is expressed as a series of integers or text strings. For example, the object name for the interfaces MIB can be expressed as 1.3.6.1.2.1.2 or iso.internet.mgmt.mib-2.interfaces. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. The following types are valid: • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE

get-type	Specifies the type of SNMP get operation to apply to the object ID specified by the *oid-value* argument. • **exact** --(Optional) Retrieves the object ID specified by the *oid-value* argument. • **next** --(Optional) Retrieves the object ID that is the alphanumeric successor to the object ID specified by the *oid-value* argument.
community	Specifies the community string to access the SNMP entity.
community-string	SNMP community string. Community string functions like passwords to access the SNMP entity. The string can consist of 1 to 32 alphanumeric characters and can be set to any of the following: • **ro** --Sets the read-only access to the SNMP entity. The default value for this community string is **public**. • **rw** --Sets read-write access to the SNMP entity. The default value for this community string is **private**.
set-type	Specifies the type of object to retrieve during the SNMP set operation. To perform a set operation, you need to specify the OID, OID type, and value.

oid-type	The type of OID. The following values are valid: • **counter32** --A 32-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0. • **gauge** --A 32-bit number with a minimum value of 0. For example, the interface speed on a router is measured using a gauge object type. • **integer** --A 32-bit number used to specify a numbered type within the context of a managed object. For example, to set the operational status of a router interface, 1 represents up and 2 represents down. • **ipv4** --IP version 4 address. • **octet string** --An octet string in hexadecimal notation used to represent physical addresses. • **string** --An octet string in text notation used to represent text strings. • **unsigned32** --A 32-bit number used to represent decimal value.
oid-type-value	Integer or text string value of the OID type specified for the SNMP set operation. The valid values for each OID type are: • **counter** --Integer value in the range from 0 to 4294967295. • **gauge--** Integer value in the range from 0 to 4294967295. • **integer--** Integer value in the range from 0 to 4294967295. • **ipv4--** IPv4 address in dotted decimal notation. • **octet string** --Text string. • **string--** Text string. • **unassigned32--** Unsigned integer value in the range from 0 to 4294967295.
ipaddr	(Optional) Specifies the IP address of the SNMP entity.
ip-address	(Optional) IP address of Network Management System (NMS) from which the objects are retrieved for SNMP get and set operations.

Command Default No requests for SNMP set or get operations are sent when the EEM applet is triggered.

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.
12.4(22)T	The **set-type**, **community**,and **ipaddr** keywords were added.

Usage Guidelines The SNMP set operation sets individual variables in the SNMP entity, whereas the SNMP get operation retrieves individual variables from the SNMP entity.

The table below shows the built-in variables in which the results of SNMP get and set operations are stored.

Table 3: EEM Built-in Variables for action info Command

Built-in Variable	Description
$_info_snmp_oid	The SNMP object ID.
$_info_snmp_value	The value string of the associated SNMP data element.

Examples The following example shows how to retrieve individual variables of an object from the SNMP entity:

```
Router(config)# event manager applet
Router(config-applet)# action 1.3 info type snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type
 exact community public ipaddr 172.17.16.69
Router(config-applet)#
```

The following example shows how to set an individual variable in the SNMP entity:

```
Router(config)# event manager applet
Router(config-applet)# action 1.4 info type snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 set-type
```

```
 integer 42220 sysName.0 community public ipaddr 172.17.16.69
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.
snmp-server community	Sets the community access string to enable access to the SNMP entity.

action info type snmp trap

To send Simple Network Management Protocol (SNMP) trap requests when an Embedded Event Manager (EEM) applet is triggered, use the **action info type snmp trap**command in applet configuration mode. To disable the sending of SNMP trap requests, use the **no** form of this command.

action *label* **info type snmp trap enterprise-oid** *enterprise-oid-value* **generic-trapnum** *generic-trap-number* **specific-trapnum** *specific-trap-number* **trap-oid** *trap-oid-value* **trap-var** *trap-variable*

no action *label* **info type**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
trap	Sends SNMP trap requests.
enterprise-oid	Specifies the enterprise OID value of the object.
enterprise-oid-value	Enterprise OID value of the object generating the SNMP trap. The OID value is enterprise specific and is expressed as a series of integers or text strings.
generic-trapnum	Specifies the generic SNMP trap number.
generic-trap-number	The generic trap number. The following generic traps and trap numbers are valid: • coldStart (0) • warmStart (1) • linkDown (2) • linkUp (3) • authenticationFaliure(4) • egpNeighborLoss(5) • enterpriseSpecific (6)
specific-trapnum	Specifies the enterprise-specific trap if the generic trap number is not set to 6.
specific-trap-number	The number associated with the trap specific to an enterprise event.

trap-oid	Specifies the object identifier of the object generating the SNMP trap.
trap-oid-value	The OID value of the object generating the SNMP trap.
trap-var	Specifies the variable associated with the instance of the object generating the trap.
trap-variable	The variable value of the object generating SNMP trap.

Command Default

No SNMP trap requests are sent by default.

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

Traps are SNMP notifications that alert the SNMP manager or the NMS to a network condition. Unlike SNMP inform requests, traps do not request the receipt from the SNMP manager.

Examples

The following example shows how to send an SNMP trap request:

```
Router(config)# event manager applet
Router(config-applet)# action 1.4 info type snmp trap enterprise-oid 1.3.6.1.4.1.1
generic-trapnum 4 specific-trapnum 7 trap-oid 1.3.6.1.4.1.1.226.0.2.1 trap-var sysUpTime
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.
snmp-server enable traps	Enables all SNMP notification types available on your system.

action info type snmp var

To create a variable for a Simple Network Management Protocol (SNMP) object identifier (OID) and its value from an Embedded Event Manager (EEM) applet, use the **action info type snmp var**command in applet configuration mode. To remove the variable, use the **no** form of this command.

action *label* **info type snmp var** *variable-name* **oid** *oid-value oid-type oid-type-value*

no action *label* **info type**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
var	Specifies the SNMP variable or the object instance of the SNMP MIB object.
variable-name	Name of the SNMP variable. For example, sysDescr.0.
oid	Requests the value of the SNMP object as specified by the SNMP OID.
oid-value	Object ID value of the data element, in SNMP dotted notation. An object identifier is expressed as a series of integers or text strings. For example, the object name for the interfaces MIB can be expressed as 1.3.6.1.2.1.2 or iso.internet.mgmt.mib-2.interfaces. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. The following types are valid: • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE

oid-type	The type of OID. The following values are valid: • **counter32** --A 32-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0. • **gauge** --A 32-bit number with a minimum value of 0. For example, the interface speed on a router is measured using a gauge object type. • **integer** --A 32-bit number used to specify a numbered type within the context of a managed object. For example, to set the operational status of a router interface, 1 represents up and 2 represents down. • **ipv4** --IP version 4 address. • **octet string** --An octet string in hex notation used to represent physical addresses. • **string** --An octet string in text notation used to represent text strings. • **unsigned32** --A 32-bit number used to represent decimal value.
oid-type-value	Integer or text string value of the OID type specified for creating a variable. The valid values for each OID type are: • **counter** --Integer value in the range from 0 to 4294967295. • **gauge--** Integer value in the range from 0 to 4294967295. • **integer--** Integer value in the range from 0 to 4294967295. • **ipv4--** IPv4 address in dotted decimal notation. • **octet string** --Text string. • **string--** Text string. • **unassigned32--** Unsigned integer value in the range from 0 to 4294967295.

Command Default No variables are created by default when an EEM applet is 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

A variable is identified by its OID and its instance. The instance is generally specified by appending a .0 to its OID. For example, sysDescr.0.

Examples

The following example shows how to create a variable for an object identifier:

```
Router(config)# event manager applet
Router(config-applet)# action 1.3 info type snmp var sysDescr.0 oid
1.3.6.1.4.1.9.9.48.1.1.1.6.1 integer 4220
Router(config-applet)#
```

Related Commands

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

action multiply

To specify the action of multiplying the variable value with a specified given integer value when an Embedded Event Manager (EEM) applet is triggered, use the **action multiply** command in applet configuration mode. To remove the calculation process, use the **no** form of this command.

action *label* **multiply** [*long-integer-1*| *variable-name-1*] [*long-integer-2*| *variable-name-2*]

no action *label* **multiply**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
long-integer-1	(Optional) First integer value for the multiplication.
variable-name-1	(Optional) First variable name for the multiplication. The value stored in the multiplier variable-name must be a long integer value or else the action will fail.
long-integer-2	(Optional) Second integer value for the multiplication.
variable-name-2	(Optional) Second variable name for the multiplication. The value stored in the multiplier variable-name must be a long integer value or else the action will fail.

Command Default

If the command is not entered within applet configuration mode, the respective applet is not registered when you exit the configuration.

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

Use the **action multiply**command to multiply the value of the variable with a given integer value. All arithmetic calculations are performed as long integers without any checks for overflow. If a statement is not associated

with this applet, events are still triggered without any action or result. A warning message stating that no statements are associated with this applet is displayed at the exit time of the configuration. All the results of the **action multiply** command are stored in $_result.

To provide a consistent user interface for the customers between the Tool Command Language (Tcl) and the CLI applet-based EEM policies, the following criteria are followed:

- Event specification criteria are written in Tcl in the Tcl-based implementation.
- Event specification data are written using the CLI applet submode configuration statements in the applet-based implementation.

Some of the keywords appear to be longer than necessary or hyphenated in the applet-based implementation because the Tcl-based implementation was developed and deployed first.

To enter applet configuration mode, use the **event manager applet** *applet-name* command after entering global configuration mode. In applet configuration mode the config prompt changes to (config-applet)# . Applet configuration mode supports three types of configuration statements:

- **event** --Specifies the event criteria that causes this applet to run.
- **action** --Performs a built-in action.
- **set** --Sets an applet variable (currently _exit_status is the only variable supported).

Examples

The following example shows how to multiply the stored variable value.

```
Router(config)# event manager applet action
Router(config-applet)# action label2 multiply 23 25
```

Related Commands

Command	Description
action add	Adds the value of the variable by the given value when an EEM applet is triggered.
action divide	Divides the value of the variable by the given value when an EEM applet is triggered.
action subtract	Subtracts the value of the variable by the given value when an EEM applet is triggered.

action puts

To enable the action of printing data directly to the local tty when an Embedded Event Manager (EEM) applet is triggered, use the **action puts**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **puts [nonewline]** *string*

no action *label* **puts**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nonewline	(Optional) Suppresses the display of the new line character.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

Data is not printed to the local tty.

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

The **action puts** command applies to synchronous events. The output of this command for a synchronous applet is directly displayed to the tty, bypassing the syslog. This command defaults to the syslog for asynchronous events. The **nonewline** keyword suppresses the display of the new line character. The output of the **action puts** command for an asynchronous applet is directed to the logger.

Examples

The following example shows how to print data directly to the local tty:

```
Router(config-applet)# event manager applet puts
```

```
Router(config-applet)# event none
Router(config-applet)# action 1 regexp "(.*) (.*) (.*)" "one two three" _match _sub1
Router(config-applet)# action 2 puts "match is $_match"
Router(config-applet)# action 3 puts "submatch 1 is $_sub1"
Router# event manager run puts
match is one two three
submatch 1 is one
Router#
```

Related Commands

Command	Description
action gets	Gets input from the local tty and stores the value in the given variable.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action regexp

To match a regular expression pattern on an input string when an Embedded Event Manager (EEM) applet is triggered, use the **action regexp**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **regexp** *string-pattern string-input* [*string-match* [*string-submatch1*] [*string-submatch2*] [*string-submatch3*]]

no action *label* **regexp**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string-pattern	The sequence of characters to be used for regular expression pattern matching.
string-input	The sequence of characters to be used as input.
string-match	(Optional) The variable name to store the entire match.
string-submatch	(Optional) The variable name to store any submatches that are present. A maximum of three submatch strings can be specified.

Command Default

No regular expression patterns are matched.

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

The *string-pattern* argument is a regular expression. If some part of the string matches the pattern, it returns 1; otherwise it returns 0. The optional *string-match* and *string-submatch* arguments store the results of the match.

The table below shows the built-in variable in which the results of the **action regexp**command are stored.

Table 4: EEM Built-in Variables for action regexp Command

Built-in Variable	Description
$_regexp_result	The result of the regular expression pattern matching is stored in this variable.

Examples

The following example shows how to define a regular expression match:

```
Router(config-applet)# event manager applet regexp
Router(config-applet)# event none
Router(config-applet)# action 1 regexp "(.*) (.*) (.*)" "one two three" _match _sub1
Router(config-applet)# action 2 puts "match is $_match"
Router(config-applet)# action 3 puts "submatch 1 is $_sub1"
Router# event manager run regexp
match is one two three
submatch 1 is one
Router#
```

Related Commands

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

action set (EEM)

To set the value of a variable when an Embedded Event Manager (EEM) applet is triggered, use the **action set**command in applet configuration mode. To remove the value of an EEM applet variable, use the **no** form of this command.

action *label* **set** *variable-name variable-value*

no action *label* **set**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
variable-name	Name assigned to the variable to be set.
variable-value	Value of the variable.

Command Default

No variable value is set.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced. This command replaces the **set** (EEM) command.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action set** command to set the value of a variable when an EEM applet is triggered.

Examples

The following example shows how to set the value of a variable:

```
Router(config-applet)# event manager applet set
Router(config-applet)# event none
Router(config-applet)# action 1 set str “this is some text”
Router(config-applet)# action 2 string range “$str” 0 6
Router(config-applet)# action 3 puts “$_string_result”
Router# event manager run set
“this is”
Router#
```

Related Commands

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

action string through D Commands

action string through D Commands

action string compare

To compare two unequal strings when an Embedded Event Manager (EEM) applet is triggered, use the **action string compare**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string compare [nocase]** [**length** *integer*] *string1 string2*

no action *label* **string compare**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nocase	(Optional) Specifies case insensitive comparison.
length	(Optional) Limits the comparison to the first integer character.
integer	(Optional) Valid values for the length argument range from 1 to 4294967295.
string1	Sequence of characters.
string2	Sequence of characters.

Command Default

Unequal strings are not compared.

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

String comparisons are performed on a byte-by-byte basis from left to right. If the strings are of unequal length, the longer string is compared greater than the shorter string. The **action string compare** command forces a comparison between two unequal strings, which is followed by an integer comparison of the result of the string comparison.

When two equal strings are compared, the result is 0 and when one string sorts before the other, the result is -1. For all other comparisons the result is 1. If the strings being compared are converted to integers, the comparison is performed between the results using the **strcmp** command.

The table below shows the built-in variable in which the results of the **action string compare** command are stored.

Table 5: EEM Built-in Variables for action string compare Command

Built-in Variable	Description
$_string_result	The result of the **action string compare** command is stored in this variable.

Examples

The following example shows how to compare two unequal strings:

```
Router(config-applet)# event manager applet compare
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some $str"
Router(config-applet)# action 2 string compare nocase length 3 “contains” “$str”
```

Related Commands

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

action string equal

To verify whether or not two strings are equal when an Embedded Event Manager (EEM) applet is triggered, use the **action string equal**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string equal [nocase]** [**length** *integer*] *string1 string2*

no action *label* **string equal**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nocase	(Optional) Specifies case insensitive comparison.
length	(Optional) Specifies the length of the value to limit the comparison.
integer	(Optional) Valid values for the length argument range from 1 to 4294967295.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

Strings are not verified as equal.

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

The **action string equal** command compares two strings and returns 1 if the strings are equal. Use **nocase** for case insensitive comparison.

The table below shows the built-in variable in which the results of the **action string equal** command are stored.

Table 6: EEM Built-in Variables for action string equal Command

Built-in Variable	Description
$_string_result	The result of the **action string equal** command is stored in this variable.

Examples

The following example shows how to verify whether or not two strings are equal:

```
Router(config-applet)# event manager applet equal
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string equal "contains" "data"
```

Related Commands

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

action string first

To return the index on the first occurrence of *string1* within *string2* when an Embedded Event Manager (EEM) applet is triggered, use the **action string first** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string first** *string1 string2* [*index-value*]

no action *label* **string first**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
index-value	(Optional) The index value to start the first test. Number in the range from 0 to 4294967295.

Command Default

The index is not returned on the first occurrence of *string1* within *string2*.

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

On the first occurrence of *string1*, the index is placed in *string2*. If *string1* is not found, it returns -1.

The table below shows the built-in variable in which the results of the **action string first** command are stored.

Table 7: EEM Built-in Variables for action string first Command

Built-in Variable	Description
$_string_result	The result of the **action string first** command is stored in this variable.

Examples

The following example shows how to return the index on the first occurrence of *string1* within *string2*:

```
Router(config-applet)# event manager applet first
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string first "contains" "$str"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run first
5
Router#
```

Related Commands

Command	Description
action string last	Returns the index on the last occurrence of *string1* within *string2*.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string index

To return the characters specified at a given index value when an Embedded Event Manager (EEM) applet is triggered, use the **action string index**command in applet configuration mode. To disable this function, use the **no** form of the command.

action *label* **string index** *string* [*value*| *end*]

no action *label* **string index**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
value	(Optional) The index value. Number in the range from 0 to 4294967295. The count starts from 0.
end	(Optional) Last character of the string.

Command Default

The characters specified at a given index value are not returned.

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

The index count starts from zero. Use the *end* argument for the last character of the string.

The table below shows the built-in variable in which the **action string index** command stores the characters.

Table 8: EEM Built-in Variables for action string index Command

Built-in Variable	Description
$_string_result	The **action string index** command stores the characters in this variable.

Examples

The following example shows how to return the character specified at a given index value:

```
Router(config-applet)# event manager applet index
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is text"
Router(config-applet)# action 2 string index "$str" 8
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run index
t
Router#
```

Related Commands

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

action string last

To return the index on the last occurrence of *string1* within *string* 2 when an Embedded Event Manager (EEM) applet is triggered, use the **action string last**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string last** *string1 string2* [*index-value*]

no action *label* **string last**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
index-value	(Optional) The index value to start the last test. Number in the range from 0 to 4294967295.

Command Default

The index is not returned on the last occurrence of *string1* within *string2*.

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

On the first occurrence of *string1*, the index is placed in *string2*. If *string1* is not found, it returns -1.

The table below shows the built-in variable in which the results of the **action string last** command are stored.

Table 9: EEM Built-in Variables for action string last Command

Built-in Variable	Description
$_string_result	The result of the **action string last** command is stored in this variable.

Examples

The following example shows how to return the index on the last occurrence of *string1* within *string2*:

```
Router(config-applet)# event manager applet last
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string last "contains" "$str"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run last
5
Router#
```

Related Commands

Command	Description
action string first	Returns the index on the first occurrence of *string1* within *string2*.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string length

To return the number of characters in a string when the Embedded Event Manager (EEM) applet is triggered, use the **action string length**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string length** *string*

no action *label* **string length**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

The number of characters in a string are not returned.

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

Use the **action string length** command to specify the action of returning the number of characters in a string when an EEM applet is triggered.

The table below shows the built-in variable in which the results of the **action string length** command are stored.

Table 10: EEM Built-in Variables for action string length Command

Built-in Variable	Description
$_string_result	The result of the **action string length** command is stored in this variable.

Examples

The following example shows how to return the number of characters in a string:

```
Router(config-applet)# event manager applet length
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string length "contains"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run length
8
Router#
```

Related Commands

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

action string match

To return 1 to the $_string_result, if the string matches the pattern when an Embedded Event Manager (EEM) applet is triggered, use the **action string match** command in applet configuration mode. To disable this action, use the **no** form of this command.

action *label* **string match [nocase]** *string-pattern string*

no action *label* **string match**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nocase	(Optional) Specifies case insensitive comparison.
string-pattern	The pattern for case insensitive comparison.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

Results of the pattern matching of strings are not returned to the $_string_result.

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

When the string matches the specified pattern, the result is 1; when the pattern does not match, the result is 0.

The table below shows the built-in variable in which the results of the **action string match**command is stored.

Table 11: EEM Built-in Variables for action string match Command

Built-in Variable	Description
$_string_result	The result of the **action string match** command is stored in this variable.

Examples

The following example shows how to return 1 to the $_string_result if the string matches the pattern:

```
Router(config-applet)# event manager applet match
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is some text"
Router(config-applet)# action 2 string match "$str" "this is"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run match
1
Router#
```

Related Commands

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

action string range

To store a range of characters in a string when an Embedded Event Manager (EEM) applet is triggered, use the **action string range** command in a pplet configuration mode . To disable this function, use the **no** form of this command.

action *label* **string range** *string start-index end-index*

no action *label* **string range**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters which can be up to 4294967295. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	The starting index string value. The range is from 0 to 4294967295.
end-index	The ending index string value. The range is from 0 to 4294967295.

Command Default

A string is not stored.

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

Use the **action string range** command to specify the action of storing a range of characters in a string when an EEM applet is triggered. The *start-index* and *end-index*arguments specify the range of the string on which to operate.

The table below shows the built-in variable in which the result of the **action string range** command is stored.

Table 12: EEM Built-in Variables for action string range Command

Built-in Variable	Description
$_string_result	The result of the **action string range** command is stored in this variable.

Examples

The following example shows how to store a range of characters in a specified string:

```
Router(config)# event manager applet store
Router(config-applet)#
action 1.0 set string "This is some text"
Router(config-applet)# action 2.0 string range "$string" 0 6
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run store
this is
Router#
```

Related Commands

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

action string replace

To store a new string by replacing the range of characters in the specified string when an Embedded Event Manager (EEM) applet is triggered, use the **action string replace**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string replace** *string start-index end-index* [*new-string*]

no action *label* **string replace**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters, which can be up to 4294967295. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	The starting index string value. The range is from 0 to 4294967295.
end-index	The ending index string value. The range is from 0 to 4294967295.
new-string	(Optional) The sequence of characters that will replace the range of characters in the string.

Command Default

A string is not stored.

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

Use the **action string replace** command to get a new string by replacing specific characters in a particular string. If the value for *new-string* argument is not specified, the characters are replaced with white space.

The table below shows the built-in variable in which the result of the **action string replace** command is stored.

Table 13: EEM Built-in Variables for action string replace Command

Built-in Variable	Description
$_string_result	The result of the **action string replace** command is stored in this variable.

Examples

The following example shows how to store the new string made by replacing the specific characters in a string:

```
Router(config)# event manager applet replace
Router(config-applet)# event none
Router(config-applet)# action 1.0 set string "This is some text"
Router(config-applet)# action 2.0 string replace "$string" 0 6 "that was"
Router(config-applet)# action 3.0 puts "$_string_result"
Router (config-applet)# end
Router# event manager run replace
that was some text
Router#
```

Related Commands

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

action string tolower

To store a specific range of characters of a string in lowercase when an Embedded Event Manager (EEM) applet is triggered, use the **action string tolower**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string tolower** *string* [*start-index*] [*end-index*]

no action *label* **string tolower**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	The sequence of characters that needs to be replaced. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	(Optional) The starting index string value. The range is from 0 to 4294967295.
end-index	(Optional) The ending index string value. The range is from 0 to 4294967295.

Command Default

A string is not stored.

Command Modes

Applet configuration (applet-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

Use the **action string tolower** command to store a specific range of characters of a string in lowercase. The *start-index* and *end-index* arguments specify the range of the string on which to operate.

The table below shows the built-in variable in which the result of the **action string tolower** command is stored.

Table 14: EEM Built-in Variables for action string tolower Command

Built-in Variable	Description
$_string_result	The result of the **action string tolower** command is stored in this variable.

Examples

The following example shows how to store a range of characters in a specific string in lowercase:

```
Router(config)# event manager applet lowercase
Router(config-applet)# action 1.0 set string "This is a STRING"
Router(config-applet)# action 2.0 string tolower "$string" 11 16
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run lowercase
string
Router#
```

Related Commands

Command	Description
action string toupper	Stores a specific range of characters of a string in uppercase.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string toupper

To store a specific range of characters of a string in uppercase when an Embedded Event Manager (EEM) applet is triggered, use the **action string toupper**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string toupper** *string* [*start-index*] [*end-index*]

no action *label* **string toupper**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Specifies the sequence of characters, that needs to be replaced. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	(Optional) The starting index string value. The range is from 0 to 4294967295.
end-index	(Optional) The ending index string value. The range is from 0 to 4294967295.

Command Default

A string is not stored.

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

Use the **action string toupper** command to store a specific range of characters of a string in uppercase. The *start-index* and *end-index* arguments specify the range of the string on which to operate.

The table below shows the built-in variable in which the result of the **action string toupper** command is stored.

Table 15: EEM Built-in Variables for action string toupper Command

Built-in Variable	Description
$_string_result	The result of the **action string toupper** command is stored in this variable.

Examples

The following example shows how to store a range of characters in a specific string in uppercase:

```
Router(config)# event manager applet uppercase
Router(config-applet)# action 1.0 set string "This is a string"
Router(config-applet)# action 2.0 string toupper "$string" 11 16
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run uppercase
STRING
Router#
```

Related Commands

Command	Description
action string tolower	Stores a specific range of characters of a string in lowercase.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string trim

To trim a string when an Embedded Event Manager (EEM) applet is triggered, use the **action string trim** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string trim** *string1* [*string2*]

no action *label* **string trim**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	(Optional) Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

By default, there is no action to trim a string.

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

Use the **action string trim** command to trim the characters in a string. This command trims the characters in *string2* from both ends of *string1*. By default, *string2* corresponds to white space.

The table below shows the built-in variable in which the result of the **action string trim** command is stored.

Table 16: EEM Built-in Variables for action string trim Command

Built-in Variable	Description
$_string_result	The result of the **action string trim** command is stored in this variable.

Examples

The following example shows how to trim a string:

```
Router(config)# event manager applet trim
Router(config-applet)# action 1.0 set string "Hello How are you?Hello"
Router(config-applet)# action 2.0 string trim "$string" "Hello "
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trim
How are you?
Router#
```

Related Commands

Command	Description
action string trimleft	Trims the characters by one string from the left end of another string.
action string trimright	Trims the characters by one string from the right end of another string.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string trimleft

To trim the characters of one string from the left end of another string when an Embedded Event Manager (EEM) applet is triggered, use the **action string trimleft**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string trimleft** *string1* [*string2*]

no action *label* **string trimleft**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	(Optional) Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

By default, there is no action to trim a string.

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

Use the **action string trimleft** command to trim a string from the left end of another string. This command trims the characters specified by *string2* from the left end of *string1*. By default, *string2* corresponds to white space.

The table below shows the built-in variable in which the result of the **action string trimleft** command is stored.

Table 17: EEM Built-in Variables for action string trimleft Command

Built-in Variable	Description
$_string_result	The result of the **action string trimleft** command is stored in this variable.

Examples

The following example shows how to trim a string from the left side of another string:

```
Router(config)# event manager applet trimleft
Router(config-applet)# action 1.0 set string "Hello How are you?"
Router(config-applet)# action 2.0 string trimleft "$string" "Hello "
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trimleft
How are you?
Router#
```

Related Commands

Command	Description
action string trim	Trims a string.
action string trimright	Trims the characters by one string from the right end of another string.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string trimright

To trim the characters one string from the right end of another string when an Embedded Event Manager (EEM) applet is triggered, use the **action string trimright**command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string trimright** *string1* [*string2*]

no action *label* **string trimright**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	(Optional) Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

By default, there is no action to trim a string.

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

Use the **action string trimright** command to trim a string from the right end of another string. This command trims the characters specified by *string2* from the right end of *string1*. By default, *string2* corresponds to white space.

The table below shows the built-in variable in which the result of the **action string trimright** command is stored.

Table 18: EEM Built-in Variables for action string trimright Command

Built-in Variable	Description
$_string_result	The result of the **action string trimright** command is stored in this variable.

Examples

The following example shows how to trim a string from the right side of another string:

```
Router(config)# event manager applet trimright
Router(config-applet)# action 1.0 set string "How are you? Hello"
Router(config-applet)# action 2.0 string trim "$string" " Hello"
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trimright
How are you?
Router#
```

Related Commands

Command	Description
action string trim	Trims a string.
action string trimleft	Trims the characters by one string from the left end of another string.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action subtract

To specify the action of subtracting the value of a variable from another value, when an Embedded Event Manager (EEM) applet is triggered, use the **action subtract** command in applet configuration mode. To undo the subtract action, use the **no** form of this command.

action *label* **subtract** {*variable-name*| *long-integer*} {*variable-name*| *long-integer*}

no action *label* **subtract**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
subtract	Subtracts the value of a variable from the value of another variable.
variable-name	String value to be placed as the variable name.
long-integer	Long integer value by which another value gets subtracted.

Command Default

By default, there is no change in the value of variables configured within an EEM applet.

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

You can use this action to subtract the value of a variable from the value of another variable. The result is stored in the variable named $_result. The value of the variable must be a long integer, else the action will fail.

Examples

The following example shows how to configure an EEM applet to subtract the value of a variable from another value:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set $var1 20
Router(config-applet)#action 1.0 set $var2 10
Router(config-applet)#action 1.0 subtract $var1 $var2
Router(config-applet)#
```

Related Commands

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

action track read

To specify the action of reading the state of a tracked object when an Embedded Event Manager (EEM) applet is triggered, use the **action track read**command in applet configuration mode. To remove the **action track read** command from the configuration, use the **no** form of this command.

action *label* **track read** *object-number*

no action *label* **track read** *object-number*

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
object-number	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command.

Command Default

The state of a tracked object is not read.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

This command generates the following result variable:

- _track_state--State of the specified tracked object. The text string returned is either up or down. If the state is up, it means that the object exists and is in an up state. If the state is down, it means that the object either does not exist or is in a down state.

This command is used to help track objects using EEM. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes such as EEM use this number

to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down. The enhanced object tracking event detector publishes an EEM event when the tracked object changes.

Examples

The following example shows how to specify event criteria based on a tracked object:

```
event manager applet track-ten
 event track 10 state any
 action 1.0 track set 10 state up
 action 2.0 track read 10
```

Related Commands

Command	Description
action track set	Specifies the action of setting the state of a tracked object when an EEM applet is triggered.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
show track	Displays tracking information.
track stub	Creates a stub object to be tracked.

action track set

To specify the action of setting the state of a tracked object when an Embedded Event Manager (EEM) applet is triggered, use the **action track set**command in applet configuration mode. To remove the **action track set** command from the configuration, use the **no** form of this command.

action *label* **track set** *object-number* **state** {**up**| **down**}

no action *label* **track set** *object-number* **state** {**up**| **down**}

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
object-number	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command.
state	Specifies the state to which the tracked object will be set.
up	Specifies that the state of the tracked object will be set to up.
down	Specifies that the state of the tracked object will be set to down.

Command Default

The state of a tracked object is not set.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

This command generates the following result variable:

- _track_state--State of the specified tracked object. The text string returned is either up or down. If the state is up, it means that the object exists and is in an up state. If the state is down, it means that the object either does not exist or is in a down state.

This command is used to help track objects using EEM. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes such as EEM use this number to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down. The enhanced object tracking event detector publishes an EEM event when the tracked object changes.

Examples

The following example shows how to specify event criteria based on a tracked object:

```
event manager applet track-ten
 event track 10 state any
 action 1.0 track set 10 state up
 action 2.0 track read 10
```

Related Commands

Command	Description
action track read	Specifies the action of reading the state of a tracked object when an EEM applet is triggered.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
show track	Displays tracking information.
track stub	Creates a stub object to be tracked.

action while

To identify the beginning of a loop of a conditional block when an Embedded Event Manager (EEM) applet is triggered, use the **action while**command in a pplet configuration mode . To disable this function, use the **no** form of this command.

action *label* **while** *string-op1 operator string-op2*

no action *label* **while**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string-op1	Specifies the first operand.
operator	Value used with the *string-op1* and s*tring-op2* operands 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.
string-op2	The second operand.

Command Default

No conditional block is specified.

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

Use the **action while** command to identify the beginning of a loop conditional block. If $_variable is found within a string, it will be substituted before the expression is tested.

Examples

The following example shows how to identify the beginning of a loop of a conditional block when an EEM applet is triggered:

```
Router(config-applet)# action 1 set _i 2
Router(config-applet)# action 2 while $_i lt 10
Router(config-applet)# action 3 action syslog msg "i is $_i"
Router(config-applet)# action 4 end
```

Related Commands

Command	Description
action else	Identifies the beginning of an else block in the if/else conditional block.
action elseif	Identifies the beginning of the if/else conditional block.
action if	Identifies the beginning of an if conditional block.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

attribute (EEM)

To specify a complex event for an Embedded Event Manager (EEM) applet, use the **attribute**command in trigger applet configuration mode. To remove the attributes, use the **no** form of this command.

attribute tag *event-tag* [**occurs** *occurs-value*]

no attribute tag *event-tag* [**occurs** *occurs-value*]

Syntax Description

tag	Specifies a tag using the *event-tag* argument that can be used with the **attribute** command to associate an event.
event-tag	String that identifies the tag.
occurs	(Optional) Specifies the number of occurrences before an EEM event is triggered. If not specified, an EEM event is triggered on the first occurrence.
occurs-value	(Optional) Number in the range from 1 to 4294967295.

Command Default

No complex events are specified for an EEM applet.

Command Modes

Trigger applet configuration (config-applet-trigger)

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

In the trigger applet configuration mode, up to eight attribute statements can be specified to build a complex event. If no attribute statements are specified, the options in the trigger statement apply to the first event defined in the applet.

Examples

The following example shows how to use the **attribute** command to specify a complex events for an EEM applet. In this example, the applet is run when the **show bgp all** command and any syslog message that contains the string “COUNT” occurs within a period of 60 seconds.

```
Router(config)# event manager applet delay_50
```

```
Router(config-applet)# event
 tag 1.0 cli pattern "show bgp all" sync yes occurs 32 period 60 maxrun 60
Router(config-applet)# event
 tag 2.0 syslog pattern "COUNT"
Router(config-applet)# trigger occurs 1 delay 50
Router(config-applet-trigger)# correlate event 1.0 or event 2.0
Router(config-applet-trigger)# attribute tag 1.0 occurs 1
Router(config-applet-trigger)# attribute tag 2.0 occurs 1
Router(config-applet-trigger)# action 1.0 cli command "show memory"
Router(config-applet)# action 2.0 cli command "enable"
Router(config-applet)# action 3.0 cli command "config terminal"
Router(config-applet)# action 4.0 cli command " ip route 192.0.2.0 255.255.255.224 192.0.2.12"
Router(config-applet)# action 91.0 cli command "exit"
Router(config-applet)# action 99.0 cli command "show ip route | incl 192.0.2.5"
```

Related Commands

Command	Description
correlate	Builds a single complex event.
trigger (EEM)	Enters trigger applet configuration mode and specifies the multiple event configuration statements for an EEM applet.

description (EEM)

To describe what an Embedded Event Manager (EEM) applet does, use the **description** (EEM) command in applet configuration mode. To remove the description of an applet, use the **no** form of this command.

description *line*

no description

Syntax Description

line	A brief description of a policy, upto 240 characters.

Command Default

By default, no description is specified for an applet.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

Use this command to describe what an EEM applet does. It is valid to have applets without a description. The Description of an applet can be added in any order, before or after any other applet configuration. Configuring a new description for an applet that already has a description, overwrites the current description.

Examples

The following example shows how to add or modify the description for an EEM:

```
Router(config)# event manager applet one
Router(config-applet)# description "This applet looks for the word count in syslog messages"
Router(config-applet)# event syslog pattern
"count"
Router(config-applet)# action 1 syslog msg hi
```

Related Commands

Command	Description
show event manager policy active	Displays EEM policies that are executed.
show event manager policy available	Displays EEM policies that are available to be registered.

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 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.
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 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 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 *sssssssss* [*.mmm*] format, where *sssssssss* 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.

event mat through R Commands

event mat through R Commands

event mat

To publish an event when a mac-address is learned in the mac-address-table, use the **event mat**command in applet configuration mode. To disable the publishing of events, use the **no** form of this command.

event [**tag** *event-tag*] **mat** {**interface** {*type number*| **regexp** *interface-name*} [**mac-address** *mac-address*]| **mac-address** *mac-address* [**interface** {*type number*| **regexp** *interface-name*}]} [**maxrun** *maxruntime-number*] [**hold-down** *seconds*] [**type** {**add**| **delete**}]

no event mat

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.
mac-address	Specifies the MAC address.
mac-address	The MAC address.
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.
hold-down	(Optional) Specifies the time to delay the event processing.

seconds	(Optional) Number that represents seconds and optional milliseconds in the format sssssssss[.mmm]. The range for seconds is from 1 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
type	(Optional) Monitors the MAC address table events. You must specify one of the following options: • **add** --Monitors only MAC address table add events. • **delete** --Monitor only MAC address table delete events.

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 either interface or mac-address. If one of them is specified, the other one is optional. All the keywords can be used in any combination.

Examples

The following example shows how to publish an event when a mac-address is learned in the mac-address-table:

```
Router(config)# event manager applet mat
Router(config-applet)# event mat interface fastethernet0 hold-down 34 type delete
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 neighbor-discovery

To publish an event when a Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) cache entry changes or a interface link status changes in an Embedded Event Manager (EEM) applet, use the **event neighbor-discovery**command in applet configuration mode. To disable the action of publishing the event, use the **no** form of this command.

event [**tag** *event-tag*] **neighbor-discovery interface** {*type number*| **regexp** *interface-name*} [**maxrun** *maxruntime-number*] *event-to-monitor*

no event neighbor-discovery

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.

event-to-monitor	

Specifies the event to be monitored on the interface. You must specify one of the following values. You can specify more than one value.

- **cdp** --Triggers an event when a matching cdp event occurs. You must specify one of the following options.
  - **add**--Triggers events only when a new cdp cache entry is created in the cdp table.
  - **all**--Triggers an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry.
  - **delete**--Triggers events only when a cdp cache entry is deleted from the cdp table.
  - **update**--Triggers an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry.
- **lldp** --Triggers an event when a matching lldp event occurs. You must specify one of the following options.
  - **add**--Triggers events only when a new cdp cache entry is created in the cdp table.
  - **all**--Triggers an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry.
  - **delete**--Triggers events only when a cdp cache entry is deleted from the cdp table.
  - **update**--Triggers an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry.
- **line-event** --Triggers an event when the interface line protocol status changes.
- **link-event** --Triggers an event when the interface link status changes. You must specify one of the following options.
  - **admindown**--Monitors link admin-down events.

	• **all**--Monitors all link events. • **deleted**--Monitors link deleted events. • **down**--Monitors link down events. • **goingdown**--Monitors link going-down events. • **init**--Monitors link init events. • **reset**--Monitors link reset events. • **testing**--Monitors link testing events. • **up**--Monitors link up events.

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 interface and at least one of cdp, lldp, link-event and line-event for the event specification to be accepted. You can use interface and maxrun keywords and the event-trigger-criteria argument in any order.

Examples The following example shows how to publish an event when CDP cache entry changes:

```
Router(config)# event manager applet discovery
Router(config-applet)# event neighbhor-discovery interface fastethernet0 cdp all
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 nf

To publish an event when a NetFlow operation is triggered in an Embedded Event Manager (EEM) applet, use the **event nf** command in applet configuration mode. To disable the action of publishing an event when NetFlow operations are triggered, use the **no** form of this command.

event [**tag** *event-tag*] **nf monitor-name** *name* **event-type** {**create**| **delete**| **update**} **[exit-event-type]** {**create**| **delete**| **update**} *subevent* **field** *field-type* **entry-value** *value-string* [**exit-value** *value-string*] **entry-op** *operator-value* [**exit-op** *operator-value*] [**rate-interval** *seconds*] [**exit-rate-interval** *seconds*] [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **nf**

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.
monitor-name *name*	Specifies the name of the NetFlow monitor.
event-type	Specifies the type of event to monitor, cache or field.
create	Creates a NetFlow event.
delete	Deletes a NetFlow event.
update	Updates a NetFlow event.
exit-event-type	The event-type (create, delete, update) at which the event will be rearmed to be monitored again.
subevent	Specifies the event and its attributes to monitor. Valid values are **event1**, **event2**, **event3**, **event4**. **Note** The subevent keywords can be used alone, together, or in any combination with each other, but each keyword can be used only once.

field *field-type*	Specifies the cache or field attribute to be monitored. One of the following attributes can be specified: • **counter** {**bytes** | **packets**}--Specifies the counter fields. • **datalink** {**dot1q** | **mac**}--Specifies the datalink (layer2) fields. • **flow** {**direction** | **sampler**}--Specifies the flow identifying fields. • **interface** {**input** | **output**}--Specifies the interface fields. • **ipv4** *field-type*-- Specifies the IPv4 fields. • **ipv6** *field-type*-- IPv6 fields • **routing** *routing-attrribute* **--** Specifies the routing attributes. • **timestamp sysuptime** {**first | last**}**--**Specifies the timestamp fields. • **transport** *field-type*-- Specifies the Transport layer fields. For more information, use the question mark (?) online help function.
entry-value *value-string*	Specifies the entry value to be compared.
exit-value *string*	(Optional) Specifies the value at which the event is set to be monitored again.
rate-interval *sec*	Specifies the rate interval value in seconds. The valid range is from 1 to 4294967295.
exit-rate-interval *sec*	(Optional) Specifies the interval value for cache rate and cache entry. The valid range is from 0 to 4294967295.
entry-op	Specifies the operator used to compare the collected usage sample with the specified value. The valid values are:

operator-value	The comparison operator. Valid values are: • **eq** - Equal to • **ge** - Greater than or equal to • **gt** - Greater than • **le** - Less than or equal to • **lt** - Less than • **wc** - Wildcard
exit-op	(Optional) The operator used to compare the current event attribute value with the exit value.
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

By default, no events are published when NetFlow 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

You can use the **event nf**command to monitor the NetFlow events. Multiple events can be specified together for additional filtering on more than one event.

Examples

The following example how to configure an applet to monitor NetFlow events:

```
Router(config)# event manager applet EventNF
```

```
Router(config-applet)# event nf event-type create monitor-name mon1 event1 entry-op eq
entry-val val1 field counter bytes long rate-interval 12
Router(config-applet)#
```

Related Commands

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

event none

To specify that an Embedded Event Manager (EEM) policy is to be registered with the EEM and can be run manually, use the **event none** command in applet configuration mode. To remove the **event none** command from the configuration file, use the **no** form of this command.

event [**tag** *event-tag*] **none** [**sync** {**yes**| **no**}] **[default]** [**maxrun** *maxruntime-number*]

no event none

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.
synch	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.
default	(Optional) The time period during which the CLI event detector waits for the policy to exit (specified in sssssssss[.mmm] format, where sssssssss 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.
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
12.3(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.

Usage Guidelines EEM usually schedules and runs policies on the basis of an event specification that is contained within the policy itself. The **event none** command allows EEM to identify an EEM policy that can either be run manually or be run when an EEM applet is triggered. To run the policy, use either the **action policy** command in applet configuration mode or the **event manager run** command in global configuration mode.

Examples The following example shows how to register a policy named manual-policy to be run manually and then how to execute the policy:

```
Router(config)# event manager applet manual-policy
Router(config-applet)# event none
Router(config-applet)# exit
Router(config)# event manager run manual-policy
```

Related Commands

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

Command	Description
event manager run	Manually runs a registered EEM policy.
show event manager policy registered	Displays registered EEM policies.

event routing

To publish an event when route entries change in Routing Information Base (RIB) infrastructure, use the **event routing** command in applet configuration mode. To stop publishing events when route entries change in RIB, use the **no** form of this command.

event[**tag** *event-tag*]**routing network** *ip-address*/*length*[**ge** *ge-length*][**le** *le-length*][**protocol** *protocol-value*][**type**{**add**| **all**| **modify**| **remove**}][**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **routing**

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.
network	Specifies the network ip address and length, whose route is to be monitored.
ip-address / *length*	The ip address and length of the network to be monitored. For example, 192.0.2.4/8.
ge *ge-length*	(Optional) Specifies the minimum prefix length to be matched.
le *le-length*	(Optional) Specifies the maximum prefix length to be matched.
ne *ne-length*	(Optional) Specifies the prefix length not to be matched.
protocol	(Optional) Specifies the protocol value for the network being monitored.
protocol-value	The network protocol value. One of the following protocols can be used: **all**, **bgp**, **connected**, **eigrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **rip**, and **static**. The default is **all**.
type	(Optional) Specifies the desired policy trigger. The default is **all**.
add	Specifies that an entry is added to the routing table.
all	Specifies that a routing table entry is added, removed, or modified.

modify	Specifies that an entry in the routing table is modified.
remove	Specifies that an entry is removed from the routing table
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, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999.

Command Default

By default, no events are published when route entries change in RIB infrastructure.

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 route-entry changes are detected in a RIB infrastructure. The network IP address for the route to be monitored must be specified. Network prefixes to be matched, protocol values, and type are optional parameters.

Note Modification of an existing static route may result in a remove event followed by an add event for the old API (v1.0) or a modify event for the new API (v2.0) depending on the Cisco IOS release.

Examples

The following example shows how a specific route entries change when many parameters is monitored:

```
Router(config
)# event manager applet EventRouting
Router(config-applet)# event routing 192.0.2.4/8 protocol static type add ge 5 maxrun 56
Router(config-applet)#
```

The following example shows the output for the Cisco IOS version that uses the old routing API (v1.0):

```
Router# show event manager detector routing
No. Name     Version     Node        Type
1  routing     01.00     node0/0     RP
```

The following example shows the output for the Cisco IOS version that uses the new routing API (v2.0):

```
Router# show event manager detector routing
No. Name     Version     Node        Type
1  routing     02.00     node0/0     RP
```

Related Commands

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

event snmp

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run by sampling Simple Network Management Protocol (SNMP) object identifier values, use the **event snmp**command in applet configuration mode. To remove the SNMP event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact**| **next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value**| **increment**| **rate**} [**exit-comb** {**or**| **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value**| **increment**| **rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true**| **false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-value* [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact**| **next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value**| **increment**| **rate**} [**exit-comb** {**or**| **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value**| **increment**| **rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true**| **false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-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.
oid	Specifies the SNMP object identifier (object ID) value in the *oid-value* argument as the event criteria.
oid-value	Object ID value of the data element, in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following: • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE
get-type	Specifies the type of SNMP get operation to be applied to the object ID specified by the *oid-value* argument.

exact	Retrieves the object ID specified by the *oid-value* argument.
next	Retrieves the object ID that is the alphanumeric successor to the object ID specified by the *oid-value* argument.
entry-op	Compares the contents of the current object ID 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	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.
entry-val	Specifies the value with which the contents of the current object ID are compared to decide if an SNMP event should be raised.
entry-value	Entry object ID value of the data element.
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 and 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.
exit-comb	(Optional) Indicates the combination of exit conditions that must be met before event monitoring is reenabled.
or	(Optional) Specifies that an exit comparison operator and an exit object ID value or an exit time value must exist.
and	(Optional) Specifies that an exit comparison operator, an exit object ID value, and an exit time value must exist.
exit-op	(Optional) Compares the contents of the current object ID with the exit value using the 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 object ID are compared to decide whether the exit criteria are met.
exit-value	(Optional) Exit object ID value of the data element.
exit-type	(Optional) Specifies a type of operation to be applied to the object ID specified by the *exit-value* argument. If not specified, the value is assumed.
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 only milliseconds are used, the format is 0.mmm.
exit-event	(Optional) Indicates whether a separate exit event is to be triggered when event monitoring is enabled after an initial event is triggered.

true	(Optional) Specifies that a separate exit event is triggered.
false	(Optional) Specifies that a separate exit event is not triggered. This is the default.
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.
poll-interval	Specifies the time interval between consecutive polls.
poll-int-value	Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 1 to 4294967295. The range for milliseconds is from 0 to 999. The minimum polling interval is 1 second.
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 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.

Command Default

No EEM events are triggered on the basis of SNMP object identifier values.

Command Modes

Applet configuration (config-applet)

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.
12.3(14)T	Optional keywords to support SNMP rate-based events were 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.
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 and associated arguments were added.

Usage Guidelines

An EEM event is triggered when one of the fields specified by an SNMP object ID crosses a defined threshold. If multiple conditions exist, the SNMP event will be triggered when all the conditions are met.

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

When the **entry-op** keyword is used and there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.

When the **exit-op** keyword is used and there is a match, an event is triggered and event monitoring is reenabled.

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

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

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

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

The increment and rate types are supported only for the following OID types: INTEGER_TYPE, COUNTER_TYPE, and COUNTER_64_TYPE.

Examples

The following example shows how an EEM applet called memory-fail will run when there is an exact match on the value of a specified SNMP object ID that represents the amount of current process memory. A message saying that process memory is exhausted and noting the current available memory will be sent to syslog.

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
 lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 1.0 syslog msg "Memory exhausted; current available memory
is $_snmp_oid_val bytes"
```

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 saying 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
```

Related Commands

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

event snmp-notification

To register the event criteria for an Embedded Event Manager (EEM) applet that is run by sampling Simple Network Management Protocol (SNMP) notification, use the **event snmp-notification**command in applet configuration mode. To remove the SNMP notification event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **snmp-notification oid** *oid-string* **oid-val** *comparison-value* **op** *operator* [**maxrun** *maxruntime-number*] [**src-ip-address** *ip-address*] [**dest-ip-address** *ip-address*] [**default** *seconds*] [**direction** {**incoming**| **outgoing**}] [**msg-op** {**drop**| **send**}]

no event [**tag** *event-tag*] **snmp-notification**

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.
oid	Specifies the SNMP object identifier (object ID) values in the *oid-val* argument as the event criteria.
oid-string	Object ID value of the data element, in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following: • COUNTER_TYPE • COUNTER_64_TYPE • GAUGE_TYPE • INTEGER_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE • TIME_TICKS_TYPE
oid-val *comparison-value*	Specifies the OID comparison value.
op	Compares the contents of the current object ID with the SNMP Protocol Data Unit (PDU) 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	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.
maxrun	(Optional) Specifies the maximum runtime of the applet. 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. The default value is 20 seconds.
src-ip-address	(Optional) Specifies the source IP address where the SNMP notification trap originates. The default is all; it is set to receive SNMP notification traps from all IP addresses.
ip-address	(Optional) The source IP address.
dest-ip-address	(Optional) Specifies the destination IP address where the SNMP notifications trap is sent. The default is all; it is set to receive SNMP traps from all destination IP addresses.
dest-ip-address	(Optional) The destination IP address.
default *seconds*	(Optional) Specifies the time period during which the snmp notification event detector waits for the policy to exit. The time period is specified in sssssssss[.mmm] format, where sssssssss must be an integer representing seconds between 0 and 4294967295 and mmm must be an integer representing milliseconds between 0 and 999.

direction	(Optional) Determines the direction of the SNMP trap or inform PDU to filter. The default is incoming. **incoming** --Specifies the incoming direction of the SNMP trap or inform PDU to filter. **outgoing** --Specifies the outgoing direction of the SNMP trap or inform PDU to filter.
msg-op	(Optional) Indicates the action to be taken on the SNMP PDU, drop it or send it once the event is triggered. **drop** --Specifies to drop the messages. **send** --Specifies to send the messages.

Command Default

No EEM events are triggered on the basis of SNMP notification object identifier values.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(20)T	This command was introduced.
15.0(1)M	This command was modified. The following keywords and arguments were added: **default**, *seconds*, **direction**, **incoming**, **outgoing**, **msg-op**, **drop**, and **send**.

Usage Guidelines

The SNMP notification event detector provides the ability to intercept SNMP trap and inform messages coming into the router. An SNMP notification event is generated when an incoming SNMP trap or inform message matches specified values or crosses specified thresholds.

The SNMP and the SNMP server manager must be configured and enabled prior to the use of the snmp-notification event detector.

An EEM event is triggered when one of the fields specified by an SNMP notification object ID crosses a defined threshold. If multiple conditions exist, the SNMP notification event is triggered when all the conditions are met.

An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following:

- INTEGER_TYPE
- COUNTER_TYPE
- GAUGE_TYPE

- TIME_TICKS_TYPE
- COUNTER_64_TYPE
- OCTET_PRIM_TYPE
- OPAQUE_PRIM_TYPE

When the **op** keyword is used and there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.

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.

Examples

The following example shows how to configure the **snmp-server community** public RW and **snmp-server manager** commands before **event snmp-notification** is configured.

```
Router(config)# snmp-server community public RW
Router(config)# snmp-server manager
```

The following example shows how an EEM applet called SNMP_Notification is being registered to run an EEM script when the router receives an SNMP notification on destination IP address 192.168.1.1 for object OID 1 whose value equals 10.

```
Router(config)# event manager applet SNMP_Notification
Router(config-applet)# event snmp-notification dest-ip-address 192.168.1.1 oid 1 op eq
oid-val 10
Router(config-applet)# action 1 policy eem_script
```

The following example shows how to intercept an outgoing SNMP trap with the OID 1.3.6.1.4.1.318.2.3.3 and OID value of "UPS: Returned from battery backup power", drop the message and send out a different one.

```
Router(config)# event manager applet SNMP_Notification
Router(config-applet)# event snmp-notification dest_ip_address 192.168.1.1 oid
1.3.6.1.4.1.318.2.3.3 op eq oid-value "UPS: Returned from battery backup power" direction
outgoing msg-op drop
```

Related Commands

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

event snmp-object

To register the Simple Network Management Protocol (SNMP) object event for an Embedded Event Manager (EEM) applet that is run by sampling the SNMP object, use the **event snmp-object** command in applet configuration mode. To remove the SNMP object event criteria, use the **no** form of this command.

event snmp-object oid *oid-value* **type** *value* **sync** {**yes**| **no**} **skip** {**yes**| **no**} **istable** {**yes**| **no**} [**default** *seconds*] [**maxrun** *maxruntime-number*]

no event snmp-object

Syntax Description

oid	Specifies the SNMP object identifier (object ID).
oid-value	Object ID value of the data element in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value.
type *value*	Specifies the type of object. The following values are valid: • **counter** --A 32-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0. • **counter64** --A 64-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0. • **gauge** --A 32-bit number with a minimum value of 0. For example, the interface speed on a router is measured using a gauge object type. • **int** --A 32-bit number used to specify a numbered type within the context of a managed object. For example, to set the operational status of a router interface, 1 represents up and 2 represents down. • **ipv4** --IP version 4 address. • **octet** --An octet string in hex notation used to represent physical addresses. • **oid** --Object identifier value. • **string** --An octet string in text notation used to represent text strings. • **uint** --A 32-bit number used to represent decimal value.

sync	Specifies the SNMP and EEM policy execution. • **no** --Policy and SNMP will run asynchronously. • **yes** --Run policy and the result determines whether to run SNMP request.
skip	Mandatory if **sync** is set to **no** and should not be used if **sync** is **yes**. Specifies whether to skip CLI command execution. • **no** --CLI command should be executed. • **yes** --CLI command should not be executed.
istable	(Optional) Specifies whether the OID is a SNMP table. • **yes** --OID is an SNMP table. • **no** --IOD is not an SNMP table.
default	(Optional) The time period during which the SNMP Object event detector waits for the policy to exit.
seconds	(Optional) 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.
maxrun	(Optional) Specifies the maximum runtime of the applet.
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. The default value is 20 seconds.

Command Modes

Applet configuration (config-applet)

Command History

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

Release	Modification
15.0(1)M1	This command was modified. The **counter64** and **oid** values were added to the **type** keyword.

Usage Guidelines

Use the **event snmp-object** command to register the SNMP object event for an EEM applet that is run by sampling SNMP object.

Examples

The following example shows how to use the **event snmp-object** command:

```
Router(config)# event manager applet test
Router(config-applet)# event snmp-object
```

Related Commands

Command	Description
action syslog	Specifies the action of writing a message to syslog when an EEM applet is triggered.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

event track

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of a Cisco IOS Object Tracking subsystem report for the specified object number, use the **event track** command in applet configuration mode. To remove the report event criteria, use the **no** form of this command.

event [*label*] [**tag** *event-tag*] **track** *object-number* [**state** {**up**| **down**| **any**}] [**maxrun** *maxruntime-number*]

no event [*label*] [**tag** *event-tag*] **track** *object-number* [**state** {**up**| **down**| **any**}] [**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.
label	(Optional) Unique identifier that can be any string. If the string contains embedded blanks, enclose it in double quotation marks.
object-number	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command.
state	(Optional) Specifies that the tracked object transition will cause an event to be raised.
up	(Optional) Specifies that an event will be raised when the tracked object transitions from a down state to an up state.
down	(Optional) Specifies that an event will be raised when the tracked object transitions from an up state to a down state.
any	(Optional) Specifies that an event will be raised when the tracked object transitions to or from any state. This is the default.
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
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.4(20)T	The **tag** and **maxrun**keywords were added to support multiple event statements within an applet.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines There are two entry variables associated with this command:

- _track_number--Number of the tracked object that caused the event to be triggered.
- _track_state--State of the tracked object when the event was triggered; valid states are “up” or “down.”

This command is used to help track objects using EEM. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes such as EEM use this number to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

Examples The following example shows how to specify event criteria based on a tracked object:

```
event manager applet track-ten
 event track 10 state any
 action 1.0 track set 10 state up
 action 2.0 track read 10
```

Related Commands

Command	Description
action track read	Specifies the action of reading the state of a tracked object when an EEM applet is triggered.
action track set	Specifies the action of setting the state of a tracked object when an EEM applet is triggered.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
show track	Displays tracking information.
track stub	Creates a stub object to be tracked.

S through Z Commands

S through Z Commands

show event manager directory user

To display the directory to use for storing user library files or user-defined Embedded Event Manager (EEM) policies, use the **show event manager directory user** command in privileged EXEC mode.

show event manager directory user [**library**| **policy**]

Syntax Description

library	(Optional) User library files.
policy	(Optional) User-defined EEM policies.

Command Default

The directories for both user library and user policy files are displayed.

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 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 directory user** command to specify the directory to use for storing user library or user policy files.

Examples

The following example shows the /usr/fm_policies folder on disk 0 as the directory to use for storing EEM user library files:

```
Router# show event manager directory user library
disk0:/usr/fm_policies
```

Related Commands

Command	Description
event manager directory user	Specifies a directory to use for storing user library files or user-defined EEM policies.

show event manager environment

To display the name and value of Embedded Event Manager (EEM) environment variables, use the **show event manager environment** command in privileged EXEC mode.

show event manager environment [**all**| *variable-name*]

Syntax Description

all	(Optional) Displays information for all environment variables. This is the default.
variable-name	(Optional) Displays information about the specified environment variable.

Command Default

If no argument or keyword is specified, information for all environment variables is displayed.

Command Modes

Privileged EXEC

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 is sample output from the **show event manager environment**command:

```
Router# show event manager environment
No.  Name                          Value
1    _cron_entry                   0-59/1 0-23/1 * * 0-7
2    _show_cmd                     show version
3    _syslog_pattern               .*UPDOWN.*Ethernet1/0.*
```

```
4     _config_cmd1                  interface Ethernet1/0
5     _config_cmd2                  no shutdown
```

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

Table 19: show event manager environment Field Descriptions

Field	Description
No.	The index number assigned to the EEM environment variable.
Name	The name given to the EEM environment variable when it was created.
Value	The text content defined for the EEM environment variable when it was created.

Related Commands

Command	Description
event manager environment	Sets an EEM environment variable.

show event manager history events

To display the Embedded Event Manager (EEM) events that have been triggered, use the **show event manager history events** command in privileged EXEC mode.

show event manager history events [detailed] [**maximum** *number*]

Syntax Description

detailed	(Optional) Displays detailed information about each EEM event.
maximum	(Optional) Specifies the maximum number of events to display.
number	(Optional) Number in the range from 1 to 50. The default is 50.

Command Modes

Privileged EXEC (#)

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 output was modified to include the Job ID and Status fields.

Usage Guidelines

Use the **show event manager history events** command to track information about the EEM events that have been triggered.

Examples

The following is sample output from the **show event manager history events**command showing that two types of events, Simple Network Management Protocol (SNMP) and application, have been triggered.

```
Router# show event manager history events
No.  Time of Event            Event Type          Name
1    Fri Aug13  21:42:57 2004  snmp                applet: SAAping1
2    Fri Aug13  22:20:29 2004  snmp                applet: SAAping1
3    Wed Aug18  21:54:48 2004  snmp                applet: SAAping1
4    Wed Aug18  22:06:38 2004  snmp                applet: SAAping1
5    Wed Aug18  22:30:58 2004  snmp                applet: SAAping1
6    Wed Aug18  22:34:58 2004  snmp                applet: SAAping1
7    Wed Aug18  22:51:18 2004  snmp                applet: SAAping1
8    Wed Aug18  22:51:18 2004  application         applet: CustApp1
```

The following is sample output from the **show event manager history events**command that includes the Job ID and Status fields:

```
Router# show event manager history events
No. Job ID  Status   Time of Event             Event Type  Name
1   1       success  Thu Sep 7  02:54:04 2006  syslog      applet: two
2   2       success  Thu Sep 7  02:54:04 2006  syslog      applet: three
3   3       success  Thu Sep 7  02:54:04 2006  syslog      applet: four
4   4       abort    Thu Sep 7  02:54:04 2006  syslog      applet: five
5   5       abort    Thu Sep 7  02:54:04 2006  syslog      applet: six
6   6       abort    Thu Sep 7  02:54:04 2006  syslog      applet: seven
7   7       abort    Thu Sep 7  02:54:04 2006  syslog      applet: eight
8   8       cleared  Thu Sep 7  02:54:04 2006  syslog      applet: nine
9   9       cleared  Thu Sep 7  02:54:04 2006  syslog      applet: ten
10  10      cleared  Thu Sep 7  02:54:04 2006  syslog      applet: eleven
```

The following is sample output from the **show event manager history events**command using the detailed keyword:

```
Router# show event manager history events detailed
No. Job ID  Status   Time of Event             Event Type          Name
1   1       success  Thu Sep 7  02:54:04 2006  syslog              applet: two
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
2   2       success  Thu Sep 7  02:54:04 2006  syslog              applet: three
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
3   3      success   Thu Sep 7  02:54:04 2006  syslog              applet: four
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
4   4      abort     Thu Sep 7  02:54:04 2006  syslog              applet: five
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
5   5      abort     Thu Sep 7  02:54:04 2006  syslog              applet: six
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
6   6      abort     Thu Sep 7  02:54:04 2006  syslog              applet: seven
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
7   7      cleared   Thu Sep 7  02:54:04 2006  syslog              applet: eight
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
8   8      cleared   Thu Sep 7  02:54:04 2006  syslog              applet: nine
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
9   9      cleared   Thu Sep 7  02:54:04 2006  syslog              applet: ten
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
10  10     success   Thu Sep 7  02:54:04 2006  syslog              applet: eleven
 msg {23:13:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console}
```

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

Table 20: show event manager history events Field Descriptions

Field	Description
No.	Event number.

Field	Description
Job ID	Unique internal EEM scheduler job identification number.
Status	Policy completion status for the policy scheduled for this event. There are three possible status values: • Success--Indicates that the policy for this event completed normally. • Abort--Indicates that the policy for this event terminated abnormally. • Cleared--Indicates that the policy for this event was removed from execution using the event manager scheduler clear command.
Time of Event	Day, date, and time when the event was triggered.
Event Type	Type of event.
Name	Name of the policy that was triggered.

Related Commands

Command	Description
event manager history size	Modifies the size of the EEM history tables.
event manager scheduler clear	Clears EEM policies that are executing or pending execution.

show event manager history traps

To display the Embedded Event Manager (EEM) Simple Network Management Protocol (SNMP) traps that have been sent, use the **show event manager history traps** command in privileged EXEC mode.

show event manager history traps [**server**| **policy**]

Syntax Description

server	(Optional) Displays SNMP traps that were triggered from the EEM server.
policy	(Optional) Displays SNMP traps that were triggered from within an EEM policy.

Command Modes

Privileged EXEC

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 **show event manager history traps** command to identify whether the SNMP traps were implemented from the EEM server or from an EEM policy.

Examples

The following is sample output from the **show event manager history traps**command:

```
Router# show event manager history traps policy
No.  Time                     Trap Type           Name
1    Wed Aug18  22:30:58 2004  policy              EEM Policy Director
```

```
2    Wed Aug18  22:34:58 2004  policy             EEM Policy Director
3    Wed Aug18  22:51:18 2004  policy             EEM Policy Director
```

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

Table 21: show event manager history traps Field Descriptions

Field	Description
No.	Trap number.
Time	Date and time when the SNMP trap was implemented.
Trap Type	Type of SNMP trap.
Name	Name of the SNMP trap that was implemented.

Related Commands

Command	Description
event manager history size	Modifies the size of the EEM history tables.

show event manager metric processes

To display Embedded Event Manager (EEM) reliability metric data for Cisco IOS Software Modularity processes, use the **show event manager metric processes**command in privileged EXEC mode.

show event manager metric processes {**all**| *process-name*}

Syntax Description

all	Displays the process metric data for all Cisco IOS Software Modularity processes.
process-name	Specific process name.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(18)SXF4	This command was introduced.

Usage Guidelines

Use this command to display the reliability metric data for Cisco IOS Software Modularity processes. The system keeps a record of when processes start and end, and this data is used as the basis for reliability analysis.

The information provided by this command allows you to get availability information for a process or group of processes. A process is considered available when it is running.

Examples

The following is partial sample output from the **show event manager metric processes**command. In this partial example, the first and last entries showing the metric data for the processes on all the cards inserted in the system are displayed.

```
Router# show event manager metric processes all
=====================================
node name: node0
process name: devc-pty, instance: 1
sub_system: 0, version: 00.00.0000
--------------------------------
last event type: process start
recent start time: Fri Oct10  20:34:40 2003
recent normal end time: n/a
recent abnormal end time: n/a
number of times started: 1
number of times ended normally: 0
number of times ended abnormally: 0
most recent 10 process start times:
--------------------------
Fri Oct10  20:34:40 2003
--------------------------
most recent 10 process end times and types:
cumulative process available time: 6 hours 30 minutes 7 seconds 378 milliseconds
cumulative process unavailable time: 0 hours 0 minutes 0 seconds 0 milliseconds
```

```
process availability:  0.100000000
number of abnormal ends within the past 60 minutes (since reload): 0
number of abnormal ends within the past 24 hours (since reload): 0
number of abnormal ends within the past 30 days (since reload): 0
.
.
.
====================================
node name: node0
process name: cdp2.iosproc, instance: 1
sub_system: 0, version: 00.00.0000
--------------------------------
last event type: process start
recent start time: Fri Oct10  20:35:02 2003
recent normal end time: n/a
recent abnormal end time: n/a
number of times started: 1
number of times ended normally: 0
number of times ended abnormally: 0
most recent 10 process start times:
-------------------------
Fri Oct10  20:35:02 2003
-------------------------
most recent 10 process end times and types:

cumulative process available time: 6 hours 29 minutes 45 seconds 506 milliseconds
cumulative process unavailable time: 0 hours 0 minutes 0 seconds 0 milliseconds
process availability:  0.100000000
number of abnormal ends within the past 60 minutes (since reload): 0
number of abnormal ends within the past 24 hours (since reload): 0
number of abnormal ends within the past 30 days (since reload): 0
```

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

Table 22: show event manager metric processes Field Descriptions

Field	Description
node name	Node name.
process name	Software Modularity process name.
instance	Instance number of the Software Modularity process.
sub_system	Subsystem number.
version	Version number.

show event manager policy active

To display Embedded Event Manager (EEM) policies that are executing, use the **show event manager policy active** command in privileged EXEC mode.

show event manager policy active [**queue-type** {**applet**| **call-home**| **axp**| **script**}| **class** *class-options*| **detailed**]

Syntax Description

queue-type	(Optional) Specifies the queue type of the EEM policy.
applet	(Optional) Specifies EEM applet policy.
call-home	(Optional) Specifies EEM Call-Home policy.
axp	(Optional) Specifies EEM axp policy.
script	(Optional) Specifies EEM script policy.
class	(Optional) Specifies EEM class policy.
class-options	Specifies the EEM class policy. You can specify either one or all of the following: • *class-letter*-- The class letter assigned for the EEM policy. Letters range from A to Z. Multiple instances of class letter can be specified. • **default** --Specifies policies registered with 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.
detailed	(Optional) Specifies the detailed content of the EEM policies.

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 active** command to display the running policies.

Examples

The following is sample output from the **show event manager policy active**command that includes the priority, scheduler node, and event type fields:

```
Router# show event manager policy active
no. job id   p s    status    time of event             event type        name
1    1       N A    wait      Wed Oct8 21:45:10 2008    syslog            continue.tcl
2    12609   N A    running   Mon Oct29 20:49:42 2007   timer watchdog    loop.tcl
```

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

Table 23: show event manager policy active Field Descriptions

Field	Description
no.	Index number automatically assigned to the policy.
job id	Unique internal EEM scheduler job identification number.
p	Priority of the policy. There are four priorities: • L--Indicates that the policy is of low priority. • H--Indicates that the policy is of high priority. • N--Indicates that the policy is of normal priority. • Z--Indicates that the policy is of least priority.
s	Scheduler node of the policy. There are two nodes: • A--Indicates that the scheduler node of this policy is active. • S--Indicates that the scheduler node of this policy is standby.

Field	Description
status	Scheduling status for the policy. There are six possible status values: • pend--Indicates that the policy is awaiting execution. • runn--Indicates that the policy is executing. • exec--Indicates that the policy has completed executing and is awaiting scheduler cleanup tasks. • hold--Indicates that the policy is being held. • wait--Indicates that the policy is waiting for a new event. • continue--Indicates that the policy receives a new event and is ready to run.
time of event	Date and time when the policy was queued for execution in the EEM server.
event type	Type of event.
name	Name of the EEM policy file.

Related Commands

Command	Description
show event manager	Shows the event manager details of an EEM policy.

show event manager policy available

To display Embedded Event Manager (EEM) policies that are available to be registered, use the **show event manager policy available** command in privileged EXEC mode.

show event manager policy available [**description** [*policy-name*]| [**detailed** *policy-filename*] [**system**| **user**]]

Syntax Description

description	(Optional) Specifies a brief description of the available policy.
policy-name	(Optional) Name of the policy.
detailed	(Optional) Displays the actual sample policy for the specified *policy-filename*.
policy-filename	(Optional) Name of sample policy to be displayed.
system	(Optional) Displays all available system policies.
user	(Optional) Displays all available user policies.

Command Default

If no keyword is specified, information for all available system and user policies is displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.3(14)T	The **user** keyword was added, and 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 **detailed** keyword and the *policy-filename* argument were added, and 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(20)T	The output was modified to display bytecode scripts with a file extension of .tbc.
15.0(1)M	The command was modified. The **description** keyword and *policy-name* argument were added.

Usage Guidelines

This command is useful if you forget the exact name of a policy required for the **event manager policy** command.

The **detailed** keyword displays the actual specified sample policy. Use **description** *policy-name* to describe a policy. If *policy-name* is not specified, the output of show command displays the description of all the available policies.

In Cisco IOS Release 12.4(20)T, EEM 2.4 introduced bytecode support to allow storage of Tcl scripts in bytecode format, and the output of this command was modified to display files with a .tbc extension as well as the usual .tcl extension for Tcl scripts.

Examples

The following is sample output from the **show event manager policy available**command:

```
Router# show event manager policy available
No.  Type    Time Created                 Name
1    system  Tue Sep 12 09:41:32 2002     sl_intf_down.tcl
2    system  Tue Sep 12 09:41:32 2002     tm_cli_cmd.tcl
```

The table below describes the fields shown in the display.

Table 24: show event manager policy available Field Descriptions

Field	Description
No.	Index number automatically assigned to the policy.
Type	Indicates whether the policy is a system policy.
Time Created	Time stamp indicating the date and time when the policy file was created.
Name	Name of the EEM policy file.

The following is sample output from the **show event manager policy available**command with the **detailed** keyword and a policy name specified:

```
Router# show event manager policy available detailed tm_cli_cmd.tcl
::cisco::eem::event_register_timer cron name crontimer2 cron_entry $_cron_entry maxrun 240
#----------------------------------------------------------------
```

```
# EEM policy that will periodically execute a cli command and email the
# results to a user.
#
# July 2005, Cisco EEM team
#
# Copyright (c) 2005 by cisco Systems, Inc.
# All rights reserved.
#------------------------------------------------------------------
### The following EEM environment variables are used:
###
### _cron_entry (mandatory)            - A CRON specification that determines
###                                      when the policy will run. See the
###                                      IOS Embedded Event Manager
###                                      documentation for more information
###                                      on how to specify a cron entry.
### Example: _cron_entry                 0-59/1 0-23/1 * * 0-7
###
### _email_server (mandatory)          - A Simple Mail Transfer Protocol (SMTP)
###                                      mail server used to send e-mail.
### Example: _email_server               mailserver.customer.com
###
```

The following is sample output from the **show event manager policy available**command showing a Tcl script with a .tcl filename extension and a bytecode script with a filename extension of .tbc. This example is for a Cisco IOS Release 12.4(20)T or later image.

```
Router# show event manager policy available
No.  Type    Time Created                 Name
1    system  Tue Jun 10 09:41:32 2008     sl_intf_down.tcl
2    system  Tue Jun 10 09:41:32 2008     tm_cli_cmd.tbc
```

Related Commands

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

show event manager policy pending

To display Embedded Event Manager (EEM) policies that are pending for execution, use the **show event manager policy pending** command in privileged EXEC mode.

show event manager policy pending [**queue-type** {**applet**| **call-home**| **axp**| **script**}| **class** *class-options*| **detailed**]

Syntax Description

queue-type	(Optional) Specifies the queue type of the EEM policy.
applet	(Optional) Specifies EEM applet policy.
call-home	(Optional) Specifies EEM Call-Home policy.
axp	(Optional) Specifies EEM axp policy.
script	(Optional) Specifies EEM script policy.
class	(Optional) Specifies EEM class policy.
class-options	(Optional) Specifies the EEM policy class. You can specify either one or all of the following: • *class-letter*-- The class letter assigned for the EEM policy. Letters range from A to Z. Multiple instances of class letter can be specified. • **default** --Specifies policies registered with 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.
detailed	(Optional) Specifies the detailed content of the EEM policies.

Command Modes

Privileged EXEC (#)

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 output was modified to include the Job ID and Status fields.
12.4(22)T	This command is supported with new options to qualify the policy queues reported in the output display and provides detailed policy information.

Usage Guidelines

Pending policies are policies that are pending execution in the EEM server execution queue. When an event is triggered, the policy that is registered to handle the event is queued for execution in the EEM server. Use the **show event manager policy pending** command to display the policies in this queue and to view the policy details.

Examples

The following is sample output from the **show event manager policy pending**command:

```
Router# show event manager policy pending
no. job id   p s    status    time of event              event type     name
1    12851   N A    pend    Mon Oct29   20:51:18 2007   timer watchdog   loop.tcl
2    12868   N A    pend    Mon Oct29   20:51:24 2007   timer watchdog   loop.tcl
3    12873   N A    pend    Mon Oct29   20:51:27 2007   timer watchdog   loop.tcl
4    12907   N A    pend    Mon Oct29   20:51:41 2007   timer watchdog   loop.tcl
5    13100   N A    pend    Mon Oct29   20:52:55 2007   timer watchdog   loop.tcl
```

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

Table 25: show event manager policy pending Field Descriptions

Field	Description
no.	Index number automatically assigned to the policy.
job id	Unique internal EEM scheduler job identification number.

Field	Description
p	Priority of the policy. There are four priorities: • L--Indicates that the policy is of low priority. • H--Indicates that the policy is of high priority. • N--Indicates that the policy is of normal priority. • Z--Indicates that the policy is of least priority.
s	Scheduler node of the policy. There are two nodes: • A--Indicates that the scheduler node of this policy is active. • S--Indicates that the scheduler node of this policy is standby.
status	Scheduling status for the policy. There are six possible status values: • pend--Indicates that the policy is awaiting execution. • runn--Indicates that the policy is executing. • exec--Indicates that the policy has completed executing and is awaiting scheduler cleanup tasks. • hold--Indicates that the policy is being held. • wait--Indicates that the policy is waiting for a new event. • continue--Indicates that the policy receives a new event and is ready to run.
time of event	Date and time when the policy was queued for execution in the EEM server.
event type	Type of event.
name	Name of the EEM policy file.

Related Commands

Command	Description
show event manager	Shows the event manager details of an EEM policy.

show event manager scheduler

To display the schedule activities of the scheduled Embedded Event Manager (EEM) policies, use the **show event manager scheduler** command in privileged EXEC mode.

show event manager scheduler thread [**queue-type** {**applet**| **call-home**| **axp**| **script**} **[detailed]**]

Syntax Description

thread	Specifies the thread for the scheduler.
queue-type	(Optional) Specifies the queue type of the EEM policy.
applet	(Optional) Specifies EEM applet policy.
call-home	(Optional) Specifies EEM Call-Home policy.
axp	(Optional) Specifies EEM axp policy.
script	(Optional) Specifies EEM script policy.
detailed	(Optional) Specifies the detailed content of the EEM policies.

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 scheduler** commandto show the EEM's scheduler activities. This command shows all the EEM execution threads from the scheduler perspective and the details of the running policies.

You can specify one or all of the following options: **applet**, **call-home**, **axp**, **script**, and **detailed**.

Examples

The following is sample output from the **show event manager scheduler**command:

```
Router# show event manager scheduler thread
1 Script threads service class default
  total: 1 running: 1 idle: 0
2 Script threads service class range A-D
  total: 3 running: 0 idle: 3
```

```
3 Applet threads service class default
  total: 32 running: 0 idle: 32
4 Applet threads service class W X
  total: 5 running: 0 idle: 5
Router# show event manager scheduler script thread detailed
1 Script threads service class default
  total: 1 running: 1 idle: 0
1 job id: 1, pid: 215, name: continue.tcl
2 Script threads service class range A-D
  total: 3 running: 0 idle: 3
3 Applet threads service class default
  total: 32 running: 0 idle: 32
4 Applet threads service class W X
  total: 5 running: 0 idle: 5
```

Related Commands

Command	Description
show event manager	Shows the event manager details of an EEM policy.

track stub-object

To create a stub object that can be tracked by Embedded Event Manager (EEM) and to enter tracking configuration mode, use the **track stub-object** command in global configuration mode. To remove the stub object, use the **no** form of this command.

track *object-number* **stub-object**

no track *object-number* **stub-object**

Syntax Description

object-number	Object number that represents the object to be tracked. The range is from 1 to 1000.

Command Default

No stub objects are created.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.
15.1(3)T	This command was modified. The valid range of the *object-number* argument increased to 1000.
15.1(1)S	This command was modified. The valid range for the *object-number* argument increased to 1000.

Usage Guidelines

Use the **track stub-object** command to create a stub object, which is an object that can be tracked and manipulated by an external process, EEM. After the stub object is created, the **default-state** command can be used to set the default state of the stub object.

EEM is a distributed, scalable, and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs.

As of Cisco IOS Release 15.1(3)T, a maximum of 1000 objects can be tracked. Although 1000 tracked objects can be configured, each tracked object uses CPU resources. The amount of available CPU resources on a router is dependent upon variables such as traffic load and how other protocols are configured and run. The ability to use 1000 tracked objects is dependent upon the available CPU. Testing should be conducted on site to ensure that the service works under the specific site traffic conditions.

Examples

The following example shows how to create and configure stub object 1 with a default state of up:

```
Router(config)#
track 1 stub-object
Router(config-track)#
 default-state up
```

Related Commands

Command	Description
default-state	Sets the default state for a stub object.
show track	Displays tracking information.

trigger (EEM)

To enter trigger applet configuration mode and specify the multiple event configuration statements for an Embedded Event Manager (EEM) applet, use the **trigger**command in applet configuration mode. To disable the multiple event configuration statements, use the **no** form of this command.

trigger [**occurs** *occurs-value*] [**period** *period-value*] [**period-start** *period-start-value*] [**delay** *delay-value*]

no trigger [**occurs** *occurs-value*] [**period** *period-value*] [**period-start** *period-start-value*] [**delay** *delay-value*]

Syntax Description

occurs	(Optional) Specifies the number of times the total correlation occurs before an EEM event is raised. When a number is not specified, an EEM event is raised on the first occurrence.
occurs-value	(Optional) Number in the range from 1 to 4294967295.
period	(Optional) Specifies the time interval during which the one or more occurrences must take place. If not specified, the time-period check is not applied.
period-value	(Optional) 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.
period-start	(Optional) Specifies the start of an event correlation window. If not specified, event monitoring is enabled after the first CRON period occurs.
period-start-value	(Optional) String that specifies the beginning of an event correlation window.
delay	(Optional) Specifies the number of seconds after which an event will be raised if all the conditions are true. If not specified, the event will be raised immediately.
delay-value	(Optional) 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 Disabled.

Command Modes Applet configuration (config-applet)

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 **trigger** command relates multiple event statements using the optional **tag** keyword with the *event-tag* argument specified in each event statement.

Examples The following example shows how to use the **trigger** command to enter trigger applet configuration mode and specify multiple event configuration statements for an EEM applet. In this example, the applet is run when the **show bgp all** command and any syslog message that contains the string "COUNT" occurs within a period of 60 seconds.

```
Router(config)# event manager applet delay_50
Router(config-applet)# event
 tag 1.0 cli pattern "show bgp all" sync yes occurs 32 period 60 maxrun 60
Router(config-applet)# event
 tag 2.0 syslog pattern "COUNT"
Router(config-applet)# trigger occurs 1 delay 50
Router(config-applet-trigger)# correlate event 1.0 or event 2.0
Router(config-applet-trigger)# attribute tag 1.0 occurs 1
Router(config-applet-trigger)# attribute tag 2.0 occurs 1
Router(config-applet-trigger)# action 1.0 cli command "show memory"
Router(config-applet)# action 2.0 cli command "enable"
Router(config-applet)# action 3.0 cli command "config terminal"
Router(config-applet)# action 4.0 cli command " ip route 192.0.2.0 255.255.255.224 192.0.2.12"
Router(config-applet)# action 91.0 cli command "exit"
Router(config-applet)# action 99.0 cli command "show ip route | incl 192.0.2.5"
```

Related Commands

Command	Description
attribute (EEM)	Specifies a complex event for an EEM applet.
correlate	Builds a single complex event.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.