Network Management Commands

absolute

To specify an absolute time for a time-range, use the **absolute** command in time-range configuration mode. To remove the time limitation, use the **no** form of this command.

absolute [**start** *time date*] [**end** *time date*]

no absolute

Syntax Description

start *time date*	(Optional) Absolute time and date that the **permit** or **deny** statement of the associated access list starts going into effect. The *time* is expressed in 24-hour notation, in the form of *hours*:*minutes*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m. The *date* is expressed in the format *day month year*. The minimum start is 00:00 1 January 1993. If no start time and date are specified, the **permit** or **deny** statement is in effect immediately.
end *time date*	(Optional) Absolute time and date that the **permit** or **deny** statement of the associated access list is no longer in effect. Same *time* and *date* format as described for the **start** keyword. The end time and date must be after the start time and date. The maximum end time is 23:59 31 December 2035. If no end time and date are specified, the associated **permit** or **deny** statement is in effect indefinitely.

Defaults

There is no absolute time when the time range is in effect.

Command Modes

Time-range configuration

Command History

Release	Modification
12.0(1)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Time ranges are used by IP and Internetwork Packet Exchange (IPX) extended access lists. For more information on using these functions, see the *Cisco IOS IP Configuration Guide* and the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*. Time ranges are applied to the **permit** or **deny** statements found in these access lists.

The **absolute** command is one way to specify when a time range is in effect. Another way is to specify a periodic length of time with the **periodic** command. Use either of these commands after the **time-range** command, which enables time-range configuration mode and specifies a name for the time range. Only one **absolute** entry is allowed per **time-range** command.

If a **time-range** command has both **absolute** and **periodic** values specified, then the **periodic** items are evaluated only after the **absolute start** time is reached, and are not further evaluated after the **absolute end** time is reached.

Note All time specifications are interpreted as local time. To ensure that the time range entries take effect at the desired times, the software clock should be synchronized using the Network Time Protocol (NTP), or some other authoritative time source. For more information, refer to the "Performing Basic System Management" document on Cisco.com.

Examples

In the following example, an access list named 'northeast' references a time range named 'xyz'. The access list and time range configuration permits traffic on Ethernet interface 0, starting at noon on January 1, 2005 and going forever.

```
time-range xyz
 absolute start 12:00 1 January 2005
!
ip access-list extended northeast
 permit ip any any time-range xyz
!
interface ethernet 0
 ip access-group northeast in
```

The configuration sample permits UDP traffic until noon on December 31, 2005. After that time, UDP traffic is no longer allowed out Ethernet interface 0.

```
time-range abc
 absolute end 12:00 31 December 2005
!
ip access-list extended northeast
 permit udp any any time-range abc
!
interface ethernet 0
 ip access-group northeast out
```

The configuration sample permits outgoing UDP traffic on Ethernet interface 0 on weekends only, from 8:00 a.m. on January 1, 2005, to 6:00 p.m. on December 31, 2006:

```
time-range weekend1
 absolute start 8:00 1 January 2005 end 18:00 31 December 2006
 periodic weekends 00:00 to 23:59
!
ip access-list extended northeast1
 permit udp any any time-range weekend1
!
interface ethernet 0
 ip access-group northeast1 out
```

Related Commands

Command	Description
deny	Sets conditions under which a packet does not pass a named access list.
periodic	Specifies a recurring (weekly) start and end time for a time range.
permit	Sets conditions under which a packet passes a named access list.
time-range	Enables time-range configuration mode and names a time range definition.

acl (WSMA)

To enable access control lists (ACLs) for restricting addresses that can connect to a Web Services Management Agent (WSMA) profile, use the **acl** command in WSMA listener configuration mode. To disable the access control lists, use the **no** form of this command.

acl *acl-number*

no acl

Syntax Description

acl-number	Defines the access control list number that can connect to the WSMA profile. Valid values are from 1 to 2799.

Command Default

The access control lists are disabled.

Command Modes

WSMA listener configuration (config-wsma-listen)

Command History

Release	Modification
12.4(24)T	This command was introduced.

Usage Guidelines

You can use the **acl** command to enable ACLs for restricting access to WSMA profiles. Use this command from the WSMA listener configuration mode. To enter this mode, enter the **wsma profile listener** command in global configuration mode.

Examples

The following example shows how to enable access control lists for a WSMA profile:

```
Router(config)# wsma profile listener prof1
Router(config-wsma-listen)# acl 34
Router(config-wsma-listen)#
```

Related Commands

Command	Description
encap	Configures an encapsulation for a WSMA profile.
idle-timeout	Sets a time for the WSMA profile to disconnect the session when there is no network traffic.
max-message	Sets the maximum size limit for incoming messages.
stealth	Disables WSMA from sending SOAP faults.
transport	Defines a transport configuration for a WSMA profile.
wsma profile listener	Configures and enables a WSMA listener profile.
wsse	Enables the WSSE for a WSMA profile.

action

To set the packet action clause, use the **action** command in VLAN access-map configuration submode. To remove an action element, use the **no** form of this command.

action {**drop** [**log**] | **forward** [**capture**] | **forward vlan** *other_vlan_ID* [**local**] | **redirect** *interface interface-number* | **port-channel** *channel-id interface interface-number* | **port-channel** *channel-id* **...**}

no action {**drop** [**log**] | **forward** [**capture**] | **forward vlan** *other_vlan_ID* [**local**] | **redirect** *interface interface-number* | **port-channel** *channel-id interface interface-number* | **port-channel** *channel-id* **...**}

Syntax Description

drop	Drops the packets.
log	(Optional) Logs the dropped packets in the software.
forward	Forwards (switched by hardware) the packets to its destination.
capture	(Optional) Sets the capture bit for the forwarded packets so that ports with the capture function enabled also receive the packets.
forward vlan *other_vlan_ID*	Forwards the packets to another VLAN. This feature is MAC Policy-Based Forwarding (MAC PBF).
local	(Optional) Allows local communication by the host. By default, PBF-specified devices on the same VLAN cannot communicate with each other.
redirect *interface*	Redirects packets to the specified interfaces; possible valid values are **ethernet**, **fastethernet**, **gigabitethernet**, and **tengigabitethernet**. See the "Usage Guidelines" section for additional valid values.
interface-number	Module and port number; see the "Usage Guidelines" section for valid values.
port-channel *channel-id*	Port channel to redirect traffic; valid values are a maximum of 64 values ranging from 1 to 256.

Defaults

This command has no default settings.

Command Modes

VLAN access-map configuration submode

Command History

Release	Modification
12.2(14)SX	Support for this command was introduced on the Supervisor Engine 720.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to the 12.2(17d)SXB release.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2SXI	Added MAC Policy-Based Forwarding (MAC PBF), using the new **forward vlan** option.

Usage Guidelines

The valid values for *interface* include the **ge-wan**, **atm**, and **pos** keywords that are supported on Catalyst 6500 series switches that are configured with a Supervisor Engine 2.

The *interface-number* argument designates the module and port number. Valid values for *interface-number* depend on the specified interface type and the chassis and module that are used. For example, if you specify a Gigabit Ethernet interface and have a 48-port 10/100BASE-T Ethernet module that is installed in a 13-slot chassis, valid values for the module number are from 1 to 13 and valid values for the port number are from 1 to 48.

Each redirect action allows you to specify a list of up to five destination interfaces. There is also a limit of up to 255 different interface lists that can be used by redirect actions.

The redirect action supports interface lists instead of single interfaces as shown in the following example:

```
[...] {redirect {{ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port} |
{port-channel channel-id}
```

The action clause specifies the action to be taken when a match occurs.

The forwarded packets are subject to any applied Cisco IOS ACLs. The **capture** keyword sets the capture bit in VACL-forwarded packets. Ports with the capture function enabled can receive VACL-forwarded packets that have the capture bit set. Only VACL-forwarded packets that have the capture bit set can be captured.

When the **log** keyword is specified, dropped packets are logged in the software. Only dropped IP packets can be logged. The **redirect** keyword allows you to specify up to five interfaces, which can be physical interfaces or EtherChannels. An EtherChannel member is not allowed to be a redirect interface.

VACLs on WAN interfaces support only the **action forward capture** command.

The action clause in a VACL can be forward, drop, capture, or redirect. Traffic can also be logged. VACLs applied to WAN interfaces do not support the redirect or log actions.

The redirect interface must be in the VLAN for which the VACL map is configured.

In a VLAN access map, if at least one ACL is configured for a packet type (IP, IPX, or MAC), the default action for the packet type is **drop** (deny).

If an ACL is not configured for a packet type, the default action for the packet type is **forward** (permit).

If an ACL for a packet type is configured and the ACL is empty or undefined, the configured action will be applied to the packet type.

By default, MAC PBF hosts in the same VLAN cannot communicate with each other. To allow local communication, use the **local** keyword.

Examples

This example shows how to define a drop and log action:

```
Router(config-access-map)# action drop log
Router(config-access-map)#
```

This example shows how to define a forward action:

```
Router(config-access-map)# action forward
Router(config-access-map)#
```

Related Commands

Command	Description
match	Specifies the match clause by selecting one or more ACLs for a VLAN access-map sequence.
show vlan access-map	Displays the contents of a VLAN-access map.
show vlan mac-pbf config	Displays details and statistics of the configured MAC PBF paths.
vlan access-map	Creates a VLAN access map or enter the VLAN access-map command mode.

action (event)

To set an action for an event, use the **action** command in event configuration mode. To disable the action for an event, use the **no** form of this command.

action {**set** | **notification**}

no action {**set** | **notification**}

Syntax Description

set	Specifies the action to set for an event.
notification	Enables notifications for events.

Command Default

No action is set for an event by default.

Command Modes

Event configuration (config-event)

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.
Cisco IOS XE Release 3.1S	This command was integrated into Cisco IOS XE Release 3.1S.

Usage Guidelines

While configuring a set of actions for an event, you can specify the object identifier of the object. You can also configure events to perform activities such as sending notifications or setting a MIB object whenever an event is triggered. If notifications are enabled for an event, the system sends a notification to the SNMP manager whenever the object configured for that event is modified.

Examples

The following example shows how to enable notifications for an event:

```
Router(config)# snmp mib event owner owner1 name test
Router(config-event)# action notification
Router(config-event-action-notification)#
```

Related Commands

Command	Description
object id	Specifies the object identifier of an object.
snmp mib event owner	Specifies the event owner for a management event.
value	Specifies a value for the object configured for an event.
wildcard	Specifies whether the object identifier can be fully specified or wildcarded during event configuration.

action add

To add the values of two variables when 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.
long-integer	Long integer value to be added to a variable.
variable-name	String value that identifies the variable name.

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 EEM and enters applet configuration mode.

action append

To append 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* **append**

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	String value that identifies the variable name.
variable-value	(Optional) Long integer value to be appended to the 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 EEM and enters applet configuration mode.

action break

To cause an immediate exit 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.

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 EEM 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.
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.

Table 1 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 cns-event

To specify the action of sending a message to the CNS Event Bus when an Embedded Event Manager (EEM) applet is triggered, use the **action cns-event** command in applet configuration mode. To remove the action of sending a message to the CNS Event Bus, use the **no** form of this command.

action *label* **cns-event msg** *msg-text*

no action *label* **cns-event msg** *msg-text*

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.
msg	Specifies the message to be sent to the CNS Event Bus.
msg-text	Character text, an environment variable, or a combination of the two. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

No messages are sent to the CNS Event Bus.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.

Examples

The following example shows how to specify a message to be sent to the CNS Event Bus when the memory-fail applet is triggered:

```
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 cns-event msg "Memory exhausted; current available
memory is $_snmp_oid_val bytes"
```

Related Commands

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

action comment

To add 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.
string	Series of characters, including embedded spaces that identifies 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 EEM and enters applet configuration mode.

action context retrieve

To retrieve 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.
key *key-name*	Provides the context name key.
variable *variable-name-pattern*	Provides a 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
action context save	Saves information across multiple policy triggers.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action context save

To save 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 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.
key *key-name*	Provides the context name key.
variable *variable-name-pattern*	Provides description of the variable.

Command Default

By default, no information is saved across multiple policy triggers.

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

action continue

To continue with a loop of actions when an Embedded Event Manager (EEM) applet is triggered, use the **action continue** command in applet configuration mode. To stop the continue action, use the **no** form of this command.

action *label* **continue**

no action *label* **continue**

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

By default, there is no 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.

Examples

The following example shows how to configure an EEM applet to continue with 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 continue
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 counter

To specify the action of setting or modifying a named counter when an Embedded Event Manager (EEM) applet is triggered, use the **action counter** command in applet configuration mode. To restore the default value to the counter, use the **no** form of this command.

action *label* **counter name** *counter-name* **value** *counter-value* **op** {**dec** | **inc** | **nop** | **set**}

no action *label* **counter name** *counter-name* **value** *counter-value* **op** {**dec** | **inc** | **nop** | **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.
name	Specifies the name of the counter to be set or modified.
counter-name	Name of the counter to be set or modified. The counter name is referenced in a registered counter type policy.
value	Specifies the value to be used to set or modify the counter.
counter-value	Number in the range from –2147483648 to 2147483647, inclusive.
op	Indicates the operator to be used with the *counter-value* to set or modify the specified counter.
dec	Specifies that the counter is decreased in value by the amount specified in the *counter-value* argument.
inc	Specifies that the counter is increased in value by the amount specified in the *counter-value* argument.
nop	Specifies that the counter value is read from the environment variable $_counter_value_remain.
set	Specifies that the counter is set to the value specified in the *counter-value* argument.

Command Default

No counter values are set or modified.

Command Modes

Applet 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.

Usage Guidelines

Use the **action counter** command when an event occurs periodically and you want an action to be implemented after a specified number of occurrences of that event. When the **action counter** command completes, an environment variable is updated as shown in Table 2.

Table 2 shows the built-in variable that is set when the **action counter** command is run.

Table 2 EEM Built-in Variables for action counter Command

Built-in Variable	Description
$_counter_value_remain	The value of the counter after the execution of the **action counter** command.

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

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

```
Router(config)# event manager applet EventCounter_A
Router(config-applet)# event timer watchdog time 60.0
Router(config-applet)# action 1.0 syslog msg “EventCounter_A”
Router(config-applet)# action 2.0 counter name critical_errors value 1 op inc
Router(config-applet)# exit
```

action decrement

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

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

no action *label* **decrement**

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	String value that identifies the variable name.
long-integer	Long integer value by which the variable gets decremented.

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.

Examples

The following example shows how to configure an EEM applet to decrement 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 decrement varname 12
Router(config-applet)#
```

Related Commands

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

action divide

To divide the dividend value by the given divisor value when an Embedded Event Manager (EEM) applet is triggered, use the **action divide** command in applet configuration mode. To remove the calculation process, use the **no** form of this command.

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

no action *label* **divide**

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 dividend integer value for the division.
variable-name-1	(Optional) First dividend variable name for the division. The value stored in the dividend variable name must be a long integer value or else the action will fail.
long-integer-2	(Optional) Second divisor integer value for the division.
variable-name-2	(Optional) Second divisor variable name for the division. The value stored in the divisor 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 divide** command to divide the dividend value with a given divisor 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 divide action command except the remainder value are saved in $_result. The remainder value of the divided integer is saved in $_remainder. Floating points (decimal) are not supported.

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 divide the value of the dividend by the value of the divisor.

```
Router(config)# event manager applet action
Router(config-applet)# action label2 divide 45 15
```

Related Commands

Command	Description
action add	Adds the value of the variable by the given value when an EEM applet is triggered.
action multiply	Multiplies 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 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 elseif

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

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

no action *label* **elseif**

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 command is not specified 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 elseif** command to identify the beginning of the else conditional action block in the else/if conditional action block. 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 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 identify the beginning of the elseif 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 $x lt 3
Router(config-applet)# action 3.0  puts "$x is less than 3"
Router(config-applet)# action 4.0 elseif $x lt 10
Router(config-applet)# action 5.0  puts "$x is less than 10"
Router(config-applet)# action 6.0 end
Router# event manager run action
5 is less than 10
Router#
```

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 ifgoto	Signifies the applet to jump to the given label if the condition is true 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.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

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 exit

To immediately exit from the running applet configuration when an Embedded Event Manager (EEM) applet is triggered, use the **action exit** command in applet configuration mode. To cancel the process of immediate exit from the running applet, use the **no** form of this command.

action *label* **exit** [*result*]

no action *label* **exit**

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.
result	(Optional) Parameter word for the exit result.

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 exit** command to immediately exit from the running applet configuration. 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 exit the applet configuration:

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

action force-switchover

To specify the action of switching to a secondary processor in a fully redundant environment when an Embedded Event Manager (EEM) applet is triggered, use the **action force-switchover** command in applet configuration mode. To remove the action of switching to a secondary processor, use the **no** form of this command.

action *label* **force-switchover**

no action *label* **force-switchover**

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

A switch to a secondary processor is not made.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.

Usage Guidelines

Before using the **action force-switchover** command, you must install a backup processor in the device. If the hardware is not fully redundant, the switchover action will not be performed.

Examples

The following example shows how to specify a switch to the secondary Route Processor (RP) when the memory-fail applet is triggered:

```
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 2.0 force-switchover
```

Related Commands

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

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 increment 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 increment 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.
variable-name	String value that identifies 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 or 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 EEM and enters applet configuration mode.

action info

To specify the action of obtaining system information when an Embedded Event Manager (EEM) applet is triggered, use the **action info** command in applet configuration mode. To remove the **action info** command from the configuration, use the **no** form of this command.

action *label* **info type** {**cli frequency** | **cli history** | **syslog frequency** | **syslog history** | **routername**}

no action *label*

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.
type	Specifies the type of information requested.
cli frequency	Requests information about the frequency of recent command-line interface (CLI) commands.
cli history	Requests information about the history of recent CLI commands.
syslog frequency	Requests information about the frequency of syslog messages.
syslog history	Requests information about the history of recent syslog messages.
routername	Requests the name of the specified router.

Command Default

No system information is requested.

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.

Usage Guidelines

Use the **action info** command when an event occurs and you want to request some system information.

Note In Cisco IOS Release 12.4(22)T several new **action info** commands were added and the **snmp oid** keywords were removed from this command page and added to the new **action info type snmp oid** command.

Table 3 shows the built-in variables that are set for the various **action info** keywords. The notation [1-N] represents that the built-in variable ends in a sequential number starting at 1 up to the maximum number of entries returned.

Table 3 EEM Built-in Variables for action info Command

Built-in Variable	Description
action info cli frequency	
$_info_cli_freq_num_entries	The number of CLI event entries.
$_info_cli_freq_pattern_[1-N]	A regular expression used to perform CLI command pattern matching.
$_info_cli_freq_time_sec_[1-N]	The seconds in Posix timer units since January 1, 1970, which represents the time the last CLI event was raised.
$_info_cli_freq_time_msec_[1-N]	The milliseconds in Posix timer units since January 1, 1970, which represents the time the last CLI event was raised.
$_info_cli_freq_match_count_[1-N]	The number of times that a CLI command matches the pattern specified by this CLI event specification.
$_info_cli_freq_raise_count_[1-N]	The number of times that this CLI event was raised.
$_info_cli_freq_sync_[1-N]	A "yes" means that event publish should be performed sychronously. The event detector will be notified when the Event Manager Server has completed publishing the event. The Event Manager Server will return a code that indicates whether or not the CLI command should be executed.
$_info_cli_freq_skip_[1-N]	A "yes" means that the CLI command should not be executed if the sync flag is not set.
$_info_cli_freq_occurs_[1-N]	Number of occurrences before an event is raised; if this argument is not specified an event is raised on the first occurrence.
$_info_cli_freq_period_sec_[1-N]	Number of occurrences must occur within this number of seconds in order to raise event; if not specified, does not apply.
$_info_cli_freq_period_msec_[1-N]	The number of occurrences must occur within this number of milliseconds in order to raise the event; if not specified, the period check does not apply.
action info cli history	
$_info_cli_hist_num_entries	The number of cli history entries.
$_info_cli_hist_cmd_[1-N]	The text of the CLI command.
$_info_cli_hist_time_sec_[1-N]	The time, in seconds, when the CLI command occurred.
$_info_cli_hist_time_msec_[1-N]	The time, in milliseconds, when the CLI command occurred.
action info routername	
$_info_routername	The name of the router.
action info syslog frequency	
$_info_syslog_freq_num_entries	The number of syslog entries.

Table 3 EEM Built-in Variables for action info Command (continued)

Built-in Variable	Description
$_info_syslog_freq_pattern_[1-N]	A regular expression used to perform syslog message pattern matching.
$_info_syslog_freq_time_sec_[1-N]	The seconds in Posix timer units since January 1, 1970, which represents the time the last event was raised.
$_info_syslog_freq_time_msec_[1-N]	The milliseconds in Posix timer units since January 1, 1970, which represents the time the last event was raised.
$_info_syslog_freq_match_count_[1-N]	The number of times that a syslog message matches the pattern specified by this syslog event specification since event registration.
$_info_syslog_freq_raise_count_[1-N]	The number of times that this syslog event was raised.
$_info_syslog_freq_occurs_[1-N]	The number of occurrences needed in order to raise the event; if not specified, the event is raised on the first occurrence.
$_info_syslog_freq_period_sec_[1-N]	The number of occurrences must occur within this number of Posix timer units in order to raise the event; if not specified, the period check does not apply.
$_info_syslog_freq_period_msec_[1-N]	The number of occurrences must occur within this number of Posix timer units in order to raise the event; if not specified, the period check does not apply.
action info syslog history	
$_info_syslog_hist_num_entries	The number of syslog history entries.
$_info_syslog_hist_msg_[1-N]	The text of the syslog message.
$_info_syslog_hist_time_sec_[1-N]	The seconds since January 1, 1970 which represent the time the syslog message was logged.
$_info_syslog_hist_time_msec_[1-N]	The milliseconds since January 1, 1970 which represent the time the syslog message was logged.

Examples

The following example shows how to configure an EEM applet to intercept configuration commands that attempt to access any loopback interface. The applet also performs a **no shutdown** command on the interface that is selected, and logs a message with the number of times that any "interface loopback" has been attempted. The console output is shown with the configuration because the final line displays the log message.

Note CLI commands that are issued from within a policy do not participate in CLI event pattern matching, and this prevents recursion.

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# event manager applet cli-match
Router(config-applet)# event cli pattern ".*interface Loopback.*" sync yes
Router(config-applet)# action 1.0 cli command "enable"
Router(config-applet)# action 1.1 cli command "$_cli_msg"
Router(config-applet)# action 1.2 cli command "no shutdown"
Router(config-applet)# action 1.3 info type cli frequency
Router(config-applet)# action 1.4 syslog msg "There have been
$_info_cli_freq_match_count_1 '$_info_cli_freq_pattern_1' matches."
Router(config-applet)# set 1.5 _exit_status 0
Router(config-applet)# end
Router#

00:37:30: %SYS-5-CONFIG_I: Configured from console by console

Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# interface loopback0
Router(config)#

00:37:43: %HA_EM-6-LOG: cli-match: There have been 27 '.*interface Loopback.*' matches.
```

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.

Table 4 shows the built-in variables in which the variables retrieved from the SNMP get operation are stored.

Table 4 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.
$_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.

Table 5 shows the built-in variables in which the results of SNMP get and set operations are stored.

Table 5 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 mail

To specify the action of sending a short e-mail when an Embedded Event Manager (EEM) applet is triggered, use the **action mail** command in applet configuration mode. To remove the **action mail** command from the configuration, use the **no** form of this command.

action *label* **mail server** *server-address* **to** *to-address* **from** *from-address* [**cc** *cc-address*] **subject** *subject* **body** *body-text*

no action *label* **mail server** *server-address* **to** *to-address* **from** *from-address* [**cc** *cc-address*] **subject** *subject* **body** *body-text*

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.
server	Specifies the e-mail server to be used for forwarding the e-mail. The e-mail server address can be any one of the following template formats: • username:password@host • username@host • host
server-address	An optional username and password with the fully qualified domain name of the e-mail server to be used to forward the e-mail.
to	Indicates that a recipient e-mail address is specified.
to-address	E-mail address where the e-mail is to be sent.
from	Indicates that the originating e-mail address is specified.
from-address	E-mail address from which the e-mail is sent.
cc	(Optional) Indicates that a copy e-mail address is specified.
cc-address	(Optional) E-mail address additional to the recipient listed in the *to-address* where the message is to be sent.
subject	Specifies the subject line content of the e-mail.
subject	Alphanumeric string. If the string contains embedded blanks, enclose it in double quotation marks.
body	Specifies the text content of the e-mail.
body-text	Alphanumeric string. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

No e-mails are sent.

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 *server-address* argument was modified to include an optional username and password.

Usage Guidelines

Use the **action mail** command when an event occurs about which you want to send an e-mail message, such as informing an administrator about the event.

In EEM 3.0 for Cisco IOS Release 12.4(22)T, the *server-address* argument includes an optional username and password along with the fully qualified domain name of the e-mail server to be used to forward the e-mail. The e-mail server name can be in any one of the following template formats:

- username:password@host
- username@host
- host

For example, username:123456@mailserver.cisco.com, or username@mailserver.cisco.com, or mailserver.cisco.com. If a username is supplied, the router will attempt to authenticate using the LOGIN AUTH dialog. If no username is supplied, no authentication is performed.

Examples

The following example shows how to send an e-mail when an EEM applet executes. The applet named EventInterface is triggered every time the receive_throttle counter for Fast Ethernet interface 0/0 is incremented by 5. The polling interval to check the counter is specified to run once every 90 seconds. When the applet is triggered, a syslog message and an e-mail are sent.

```
Router(config)# event manager applet EventInterface
Router(config-applet)# event interface name FastEthernet0/0 parameter receive_throttle
entry-op ge entry-val 5 entry-val-is-increment true poll-interval 90
Router(config-applet)# action 1.0 syslog msg "Applet EventInterface"
Router(config-applet)# action 1.1 mail server mailserver.cisco.com to
engineering@cisco.com from devtest@cisco.com cc manager@cisco.com subject
"Receive_throttle counter incremented" body "Receive_throttle counter for FastEthernet0/0
interface has incremented by 5"
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager 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 policy

To specify the action of manually running an Embedded Event Manager (EEM) policy when an EEM applet is triggered, use the **action policy** command in applet configuration mode. To remove the **action policy** command from the configuration, use the **no** form of this command.

action *label* **policy** *policy-filename*

no action *label* **policy** *policy-filename*

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.
policy-filename	Name of the EEM policy to be run manually. The policy must be previously registered using the **event none** command and must not be the same as the current policy.

Command Default

No EEM policies are run.

Command Modes

Applet 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.

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 be run manually or 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 policy-manual to be run manually and then to execute the policy:

```
Router(config)# event manager applet policy-manual
Router(config-applet)# event none policy-manual
Router(config-applet)# action label1 policy policy-manual
```

Related Commands

Command	Description
event manager run	Manually runs a registered EEM policy.
event none	Registers an EEM applet that is to be run manually.
show event manager policy registered	Displays registered EEM policies.

action publish-event

To specify the action of publishing an application-specific event when the event specified for an Embedded Event Manager (EEM) applet is triggered, use the **action publish-event** command in applet configuration mode. To remove the action of publishing an application-specific event, use the **no** form of this command.

action *label* **publish-event sub-system** *sub-system-id* **type** *event-type* **arg1** *argument-data* [**arg2** *argument-data*] [**arg3** *argument-data*] [**arg4** *argument-data*]

no action *label* **publish-event**

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.
sub-system	Specifies an identifier for the subsystem named in the *sub-system-id* argument that will publish the application event.
sub-system-id	Identifier of the subsystem. Number in the range from 1 to 4294967295. If the event is to be published by an EEM policy, the *sub-system-id* reserved for a customer policy is 798.
type	Specifies the value of an event type within the specified event.
event-type	Event type value. Number in the range from 1 to 4294967295.
arg1	Specifies that argument data is to be passed to the application-specific event when the event is published.
argument-data	Character text, an environment variable, or a combination of the two. Optional when used with the **arg2**, **arg3**, or **arg4** keywords.
arg2 **arg3** **arg4**	(Optional) Specifies that argument data is to be passed to the application-specific event when the event is published.

Command Default

No application-specific events are published.

Command Modes

Applet 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.

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 the argument 1 argument data passed 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 sub-system 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.

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.

Table 6 shows the built-in variable in which the results of the **action regexp** command are stored.

Table 6 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 reload

To specify the action of reloading the Cisco IOS software when an Embedded Event Manager (EEM) applet is triggered, use the **action reload** command in applet configuration mode. To remove the action of reloading the Cisco IOS software, use the **no** form of this command.

action *label* **reload**

no action *label* **reload**

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

No reload of the Cisco IOS software is performed.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.

Usage Guidelines

Before configuring the **action reload** command, you should ensure that the device is configured to reboot the software version that you are expecting. Use the **show startup-config** command and look for any **boot system** commands.

Examples

The following example shows how to reload the Cisco IOS software when the memory-fail applet is triggered:

```
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 3.0 reload
```

Related Commands

Command	Description
boot system	Configures the locations from which the router loads software when the router reboots.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
show startup-config	Displays the configuration to be run when the router reboots.

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 snmp-object-value

To set the object ID and value to be returned by the Simple Network Management Protocol (SNMP) get request when an Embedded Event Manager (EEM) applet is triggered, use the **action snmp-object-value** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **snmp-object-value** *oid-type oid-type-value* **next-oid** *oid-value*

no action *label*

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-type	The type of OID. 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. • **gauge**—A 32-bit number with a minimum value of 0. A gauge object type is used, for example, to measure the interface speed on a router. • **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, use 1 for “up” and 2 for “down”. • **ipv4**—IP version 4 address. • **octet**—An octet string in hex notation used to represent physical addresses. • **oid**—SNMP object identifier (object ID) in dotted notation. • **string**—An octet string in text notation used to represent text strings. • **uint**—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. • **counter64**—Text string. • **gauge**—Integer value in the range from 0 to 4294967295. • **int**—Integer value in the range from 0 to 4294967295. • **ipv4**—IPv4 address in dotted decimal notation. • **octet**—Text string. • **oid**—Text string. • **string**—Text string. • **uint**—Unsigned integer value in the range from 0 to 4294967295.

next-oid	Requests the value of the next 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: • COUNTER_64_TYPE • COUNTER_TYPE • GAUGE_TYPE • INTEGER_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE • TIME_TICKS_TYPE

Command Default

By default, no object ID or value is specified.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
15.0(1)M	This command was introduced.
15.1(3)T	This command was modified. The **oid** keyword was added.

Usage Guidelines

Use the **action snmp-object-value** command to set the object ID and value to be returned for the SNMP get request.

Examples

The following example shows how to set the object ID and value to be returned by the SNMP get request.

```
Router# configure terminal
Router(config)# event manager applet action
Router(config-applet)# event snmp-object oid 1.9.9.9.9 type gauge sync yes
Router(config-applet)# action 1 syslog msg "oid = $_snmp_oid"
Router(config-applet)# action 2 syslog msg "request = $_snmp_request"
Router(config-applet)# action 3 syslog msg "request_type = $_snmp_request_type"
Router(config-applet)# action 4 syslog msg "value = $_snmp_value"
Router(config-applet)# action 5 snmp-object-value gauge 1111 next-oid 1.2.3.4
Router(config-applet)# action 6 exit 1
```

Related Commands

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

action snmp-trap

To specify the action of generating a Simple Network Management Protocol (SNMP) trap when an Embedded Event Manager (EEM) applet is triggered, use the **action snmp-trap** command in applet configuration mode. To remove the action of generating an SNMP trap, use the **no** form of this command.

action *label* **snmp-trap** [**intdata1** *integer*] [**intdata2** *integer*] [**strdata** *string*]

no action *label* **snmp-trap**

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.
intdata1	(Optional) Specifies an integer to be sent in the SNMP trap message to the SNMP agent.
intdata2	(Optional) Specifies a second integer to be sent in the SNMP trap message to the SNMP agent.
integer	(Optional) Integer value.
strdata	(Optional) Specifies a string to be sent in the SNMP trap message to the SNMP agent.
string	(Optional) Sequence of up to 256 characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

No SNMP traps are generated when an EEM applet is triggered.

Command Modes

Applet 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.

Usage Guidelines

Before configuring this command, you must enable the **snmp-server enable traps event-manager** command to permit SNMP traps to be sent from the Cisco IOS device to the SNMP server. Other relevant **snmp-server** commands must also be configured.

This command generates an asynchronous message that is sent from the Cisco IOS device to the SNMP agent. The SNMP agent can be coded to understand customized data such as the optional integer and string data that can be sent in the SNMP trap message.

The SNMP trap that is generated uses the EEM MIB, CISCO-EMBEDDED-EVENT-MGR-MIB.my. Details about the MIB can be found using Cisco MIB Locator at the following URL:

http://www.cisco.com/go/mibs

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
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
snmp-server enable traps event-manager	Permits Embedded Event Manager SNMP traps to be sent from a Cisco IOS device to the SNMP server.