event manager scheduler suspend

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

event manager scheduler suspend

no event manager scheduler suspend

Syntax Description

This command has no arguments or keywords.

Command Default

Policy scheduling is active.

Command Modes

Global configuration

Command History

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

Usage Guidelines

Use the **event manager scheduler suspend** command to suspend all policy scheduling requests and do no scheduling until you enter the **no** form of the command. The **no** form of the command resumes policy scheduling and executes any pending policies.

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

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

Examples

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

```
Router(config)# event manager scheduler suspend

May 19 14:31:22.439: fm_server[12330]: %HA_EM-6-FMS_POLICY_EXEC: fh_io_msg: Policy
execution has been suspended
```

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

```
Router(config)# no event manager scheduler suspend

May 19 14:31:40.449: fm_server[12330]: %HA_EM-6-FMS_POLICY_EXEC: fh_io_msg: Policy
execution has been resumed
```

Related Commands

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

event manager session cli username

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

event manager session cli username *username*

no event manager session cli username *username*

Syntax Description

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

Command Default

No username is associated with EEM CLI sessions.

Command Modes

Global configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.

Usage Guidelines

Use the **event manager session cli username** command to assign a username for EEM policy CLI sessions when TACACS+ is used for command authorization.

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

Examples

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

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

Related Commands

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

event manager update user policy

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

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

Syntax Description

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

Command Default

No EEM policies are registered.

Command Modes

Privileged EXEC (#)

Command History

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

Usage Guidelines

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

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

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

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

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

Examples

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

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

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

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

Related Commands

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

event mat

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

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

no event mat

Syntax Description

tag	(Optional) Specifies a tag using the event-tag argument that can be used with the trigger command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
interface	Specifies the interface.
type number	Interface type and number.
regexp *interface-name*	Specifies a regular expression pattern to match against interface names.
mac-address	Specifies the MAC address.
mac-address	The MAC address.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the maxrun keyword is specified, the maxruntime-number value must be specified. If the maxrun keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.
hold-down	(Optional) Specifies the time to delay the event processing.
seconds	(Optional) Number that represents seconds and optional milliseconds in the format ssssssssss[.mmm]. The range for seconds is from 1 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
type	(Optional) Monitors the MAC address table events. You must specify one of the following options: • **add**—Monitors only MAC address table add events. • **delete**—Monitor only MAC address table delete events.

Command Default

By default, no events are published.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(52)SE	This command was introduced.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.
15.1(3)T	This command was integrated into Cisco IOS Release 15.1(3)T.

Usage Guidelines

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

Examples

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

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

Related Commands

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

event neighbor-discovery

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

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

no event neighbor-discovery

Syntax Description

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

maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.
event-to-monitor	Specifies the event to be monitored on the interface. You must specify one of the following values. You can specify more than one value. • **cdp**—Triggers an event when a matching cdp event occurs. You must specify one of the following options. – **add**—Triggers events only when a new cdp cache entry is created in the cdp table. – **all**—Triggers an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry. – **delete**—Triggers events only when a cdp cache entry is deleted from the cdp table. – **update**—Triggers an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry. • **lldp**—Triggers an event when a matching lldp event occurs. You must specify one of the following options. – **add**—Triggers events only when a new cdp cache entry is created in the cdp table. – **all**—Triggers an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry. – **delete**—Triggers events only when a cdp cache entry is deleted from the cdp table. – **update**—Triggers an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry. • **line-event**—Triggers an event when the interface line protocol status changes. • **link-event**—Triggers an event when the interface link status changes. You must specify one of the following options. – **admindown**—Monitors link admin-down events. – **all**—Monitors all link events. – **deleted**—Monitors link deleted events. – **down**—Monitors link down events. – **goingdown**—Monitors link going-down events. – **init**—Monitors link init events. – **reset**—Monitors link reset events. – **testing**—Monitors link testing events. – **up**—Monitors link up events.

Command Default

By default, no events are published.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(52)SE	This command was introduced.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.
15.1(3)T	This command was integrated into Cisco IOS Release 15.1(3)T.

Usage Guidelines

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

Examples

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

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

Related Commands

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

event nf

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

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

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

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
monitor-name *name*	Specifies the name of the NetFlow monitor.
event-type	Specifies the type of event to monitor, cache or field.
create	Creates a NetFlow event.
delete	Deletes a NetFlow event.
update	Updates a NetFlow event.
exit-event-type	The event-type (create, delete, update) at which the event will be rearmed to be monitored again.
subevent	Specifies the event and its attributes to monitor. Valid values are **event1**, **event2**, **event3**, **event4**. **Note** The subevent keywords can be used alone, together, or in any combination with each other, but each keyword can be used only once.
field *field-type*	Specifies the cache or field attribute to be monitored. One of the following attributes can be specified: • **counter** {**bytes** | **packets**}—Specifies the counter fields. • **datalink** {**dot1q** | **mac**}—Specifies the datalink (layer2) fields. • **flow** {**direction** | **sampler**}—Specifies the flow identifying fields. • **interface** {**input** | **output**}—Specifies the interface fields. • **ipv4** *field-type*—Specifies the IPv4 fields. • **ipv6** *field-type*—IPv6 fields • **routing** *routing-attrribute*—Specifies the routing attributes. • **timestamp sysuptime** {**first** | **last**}—Specifies the timestamp fields. • **transport** *field-type*—Specifies the Transport layer fields. For more information, use the question mark (?) online help function.
entry-value *value-string*	Specifies the entry value to be compared.
exit-value *string*	(Optional) Specifies the value at which the event is set to be monitored again.

rate-interval *sec*	Specifies the rate interval value in seconds. The valid range is from 1 to 4294967295.
exit-rate-interval *sec*	(Optional) Specifies the interval value for cache rate and cache entry. The valid range is from 0 to 4294967295.
entry-op	Specifies the operator used to compare the collected usage sample with the specified value. The valid values are:
operator-value	The comparison operator. Valid values are: • **eq** - Equal to • **ge** - Greater than or equal to • **gt** - Greater than • **le** - Less than or equal to • **lt** - Less than • **wc** - Wildcard
exit-op	(Optional) The operator used to compare the current event attribute value with the exit value.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.

Command Default

By default, no events are published when NetFlow operations are triggered.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

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

Examples

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

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

Related Commands

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

event none

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

event [**tag** *event-tag*] **none** [**maxrun** *maxruntime-number*]

no event none

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

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

Command Modes

Applet configuration (config-applet).

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.

Usage Guidelines

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

Examples

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

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

Related Commands

Command	Description
action policy	Registers an EEM policy with EEM.
event manager applet	Registers an EEM applet with EEM and enters applet configuration mode.
event manager run	Manually runs a registered EEM policy.
show event manager policy registered	Displays registered EEM policies.

event oir

To specify that an Embedded Event Manager (EEM) applet be run on the basis of an event raised when a hardware card online insertion and removal (OIR) occurs, use the **event oir** command in applet configuration mode. To remove the **event oir** command from the configuration, use the **no** form of this command.

event [***tag*** *event-tag*] **oir** [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **oir** [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM applets are run on the basis of an OIR event.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.

Examples

The following example shows how to configure an EEM applet to be run on the basis of an OIR event:

```
Router(config)# event manager applet oir-event
Router(config-applet)# event oir
Router(config-applet)# exit
```

Related Commands

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

event owner

To specify the event owner for an event trigger according to the trigger type and status of the trigger, use the **event owner** command in event trigger existence or event trigger boolean configuration mode. To disable configuration and set the default parameters, use the **no** form of this command.

event owner *event-owner* **name** *event-name*

no event owner

Syntax Description

event-owner	Owner of the event.
name	Indicates the name of the event.
event-name	Unique name of the event.

Command Default

By default, the event owner and event name are not specified.

Command Modes

Event trigger existence configuration (config-event-trigger-existence)
Event trigger boolean configuration (config-event-trigger-boolean)

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

The event is identified by the *event-owner* and *event-name* values and is configured by using the **snmp mib event** command. Events are enabled by using the **enable** command.

Examples

The following example shows how to specify event owner for the existence trigger test:

```
Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test existence
Router(config-event-trigger-existence)# event owner jim name event2
Router(config-event-trigger-existence)#
```

The following example shows how to specify event owner for the Boolean trigger test:

```
Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test boolean
Router(config-event-trigger-boolean)# event owner jane name event2
Router(config-event-trigger-boolean)#
```

Related Commands

Command	Description
snmp mib event trigger owner	Specifies an event trigger owner while configuring management event trigger information.
test boolean	Configures parameters for the Boolean trigger test.
test existence	Configures parameters for the existence trigger test.

event process

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of an event raised when a Cisco IOS Software Modularity process starts, restarts, or stops, use the **event process** command in applet configuration mode. To remove the process event criteria, use the **no** form of this command.

event process {**abort** | **start** | **term** | **user-restart** | **user-shutdown**} **path** *process-name* [**instance** *instance-value*] [**node** *node-value*]

no event process {**abort** | **start** | **term** | **user-restart** | **user-shutdown**} **path** *process-name* [**instance** *instance-value*] [**node** *node-value*]

Syntax Description

abort	Specifies that an event is triggered when the specified process aborts with one of the following abnormal conditions: • A nonzero exit status. • A kernel-generated signal is received. • A SIGTERM or SIGKILL signal is received but not as a result of a user request.
start	Specifies that an event is triggered when the specified process is started.
term	Specifies that an event is triggered when the specified process stops normally.
user-restart	Specifies that an event is triggered when there is a process restart request from the CLI command.
user-shutdown	Specifies that an event is triggered when there is a process stop request.
path *process-name*	Specifies the path of the process including the process name. If the value of the *process-name* argument contains embedded blanks, enclose it in double quotation marks.
instance *instance-value*	(Optional) Specifies the process instance ID. The ID must be a number in the range of 1 to 4294967295.
node *node-value*	(Optional) Specifies the node name which is a concatenation of the hardware slot number and the hardware CPU number.

Command Default

No EEM events are triggered on the basis of a Cisco IOS Software Modularity process starting, restarting, or stopping.

Command Modes

Applet configuration (config-applet)

Command History

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

Examples

The following example shows how to specify that an event is triggered when a Software Modularity process starts:

```
Router(config)# event manager applet process_term
Router(config-applet)# event process start path “cdp2.iosproc”
```

Related Commands

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

event resource

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of an Embedded Resource Manager (ERM) event report for a specified policy, use the **event resource** command in applet configuration mode. To remove the report event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **resource policy** *policy-filename* [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **resource policy** *policy-filename* [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
policy	Indicates that a specific policy is identified.
policy-filename	Policy name.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM event criteria are specified.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.4(20)T	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

The resource event detector publishes an event when the ERM reports an event for the specified policy. The ERM infrastructure tracks resource depletion and resource dependencies across processes and within a system to handle various error conditions. The error conditions are handled by providing an equitable sharing of resources between various applications. The ERM framework provides a communication mechanism for resource entities and allows communication between these resource entities from numerous locations. The ERM framework also helps in debugging the CPU and memory-related issues. The ERM monitors system resource usage to better understand scalability needs by allowing you to configure threshold values for resources such as CPU, buffer, and memory.

Examples

The following example shows how to specify event criteria based on an ERM event report for a policy defined to report high CPU usage:

```
Router(config)# event manager applet policy-one
Router(config-applet)# event resource policy cpu-high
Router(config-applet)# action 1.0 syslog msg "CPU high at $_resource_current_value 
percent"
```

Related Commands

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

event rf

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of Redundancy Framework (RF) state change notifications, use the **event rf** command in applet configuration mode. To remove the RF event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **rf event** *rf-state-name* [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **rf event** *rf-state-name* [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
event	Compares the regular expression contained in the *rf-state-name* argument with an RF state change notification. If there is a match, an event is triggered. The *rf-state-name* argument takes one of the following values: • RF_EVENT_CLIENT_PROGRESSION • RF_EVENT_CONTINUE_PROGRESSION • RF_EVENT_GO_ACTIVE • RF_EVENT_GO_ACTIVE_EXTRALOAD • RF_EVENT_GO_ACTIVE_HANDBACK • RF_EVENT_GO_STANDBY • RF_EVENT_KEEP_ALIVE • RF_EVENT_KEEP_ALIVE_TMO • RF_EVENT_LOCAL_PROG_DONE • RF_EVENT_NEGOTIATE • RF_EVENT_NOTIFICATION_TMO • RF_EVENT_PEER_PROG_DONE • RF_EVENT_STANDBY_PROGRESSION • RF_EVENT_START_PROGRESSION • RF_EVENT_SWACT_INHIBIT_TMO • RF_PROG_ACTIVE • RF_PROG_ACTIVE_DRAIN • RF_PROG_ACTIVE_FAST • RF_PROG_ACTIVE_POSTCONFIG • RF_PROG_ACTIVE_PRECONFIG • RF_PROG_EXTRALOAD • RF_PROG_HANDBACK • RF_PROG_INITIALIZATION • RF_PROG_PLATFORM_SYNC

	• RF_PROG_STANDBY_BULK • RF_PROG_STANDBY_COLD • RF_PROG_STANDBY_CONFIG • RF_PROG_STANDBY_FILESYS • RF_PROG_STANDBY_HOT • RF_REGISTRATION_STATUS • RF_STATUS_MAINTENANCE_ENABLE • RF_STATUS_MANUAL_SWACT • RF_STATUS_OPER_REDUNDANCY_MODE_CHANGE • RF_STATUS_PEER_COMM • RF_STATUS_PEER_PRESENCE • RF_STATUS_REDUNDANCY_MODE_CHANGE • RF_STATUS_SWACT_INHIBIT
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm]format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999.

Command Default

No EEM events are triggered.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.4(20)T	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

An EEM event is triggered when the expression in the *rf-state-name* argument matches an RF state change notification. The RF event detector publishes an event when one or more RF events occur during synchronization in a dual Route Processor (RP) system.

Examples

The following example shows how to specify event criteria based on an RF state change notification:

```
Router(config)# event manager applet start-rf
Router(config-applet)# event rf event rf_prog_initialization
Router(config-applet)# action 1.0 syslog msg "rf state rf_prog_initialization reached"
```

Related Commands

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

event routing

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

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

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

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
network	Specifies the network ip address and length, whose route is to be monitored.
ip-address/length	The ip address and length of the network which needs to be monitored. Example, 192.0.2.4/8.
ge *ge-length*	(Optional) Specifies the minimum prefix length to be matched.
le *le-length*	(Optional) Specifies the maximum prefix length to be matched.
ne *ne-length*	(Optional) Specifies the prefix length not to be matched.
protocol	(Optional) Specifies the protocol value for the network being monitored.
protocol-value	The network protocol value. One of the following protocols: **all**, **bgp**, **connected**, **eigrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **rip**, **static** can be used. The default is **all**.
type	(Optional) Specifies the desired policy trigger. The default is **all**.
add	Specifies that an entry is added to the routing table.
all	Specifies that a routing table entry is added, removed or modified.
modify	Specifies that an entry in the routing table is modified.
remove	Specfies that an entry is removed from the routing table
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

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

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

An EEM event is published when route-entry changes are detected in a RIB infrstructure. The network ip address whose route needs to be monitored must be specified. Network prefixes to be matched, protocol values and type are optional parameters.

Note Modification of an existing static route may result in a remove event followed by an add event for the old API (V1.0) or a modify event for the new API (V2.0) depending on the IOS release.

Examples

The following example shows how a specific route when many parameters are monitored:

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

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

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

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

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

Related Commands

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

event rpc

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

event [**tag** *event-tag*] **rpc** [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **rpc** [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

Event RPC is disabled.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

Use this command to allow an EEM applet to be executed using the RPC command.

When an applet is configured to run using the RPC command, all output is still sent to the Command-Line-Interface (CLI) console and not sent in the RPC reply message; so an RPC ED applet is not very useful.

Examples

The following example shows how to configure the **event rpc** command with maximum runtime set to 3000 seconds.

```
Router(onfig-applet)# event rpc maxrun 3000
```

The following example shows how to send a Simple Object Access Protocol (SOAP) message request and receive reply:

Request Syntax

```
<?xml version="1.0"?>
<SOAP:Envelope xmlns:SOAP="http://www.cisco.com/eem.xsd">
<SOAP:Body>
  <run_emscript>
    <script_name> name of script </script_name>
    <argc> argc value </argc>
    <arglist>
    <l> argv1 value </l>
    <l> argv2 value </l>
          …
    <l> argvn value </l>
    </arglist>
  </run_Eemscript>
</SOAP:Body>
</SOAP:Envelope>
```

To send the request, enter "]]>]]>"

Reply Syntax

```
<?xml version="1.0"?>
<SOAP:Envelope xmlns:SOAP="http://www.cisco.com/eem.xsd">
<SOAP:Body>
   <run_Eemscript_response>
      <return_code> rc </return_code>
      <output> output string </output>
   </run_eemscript_response>
</SOAP:Body>
</SOAP:Envelope>
```

The following example shows how to configure the applet called RPC_example:

```
event manager applet RPC_example
 event rpc
 action output puts "hello world"
```

The following example shows how to run the applet using SSH:

```
infra-view11 {/users/johndoe} ssh -2 -s user@172.16.0.0 eem_rpc
Password:


<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://www.cisco.com/eem.xsd">
  <SOAP:Body>
    <run_eemscript>
      <script_name>RPC_example</script_name>
    </run_eemscript>
  </SOAP:Body>
</SOAP:Envelope>

]]>]]>
<?xml version="1.0" encoding="UTF-8"?><SOAP:Envelope
xmlns:SOAP="http://www.cisco.com/eem.xsd"><SOAP:Body><run_eemscript_response><return_code>
0</return_code><output></output></run_eemscript_response></SOAP:Body></SOAP:Envelope>]]>]]
>
```

Related Commands

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

event snmp

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

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

no event [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact** | **next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value** | **increment** | **rate**} [**exit-comb** {**or** | **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value** | **increment** | **rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true** | **false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-value* [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
oid	Specifies the SNMP object identifier (object ID) value in the *oid-value* argument as the event criteria.
oid-value	Object ID value of the data element, in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following: • INTEGER_TYPE • COUNTER_TYPE • GAUGE_TYPE • TIME_TICKS_TYPE • COUNTER_64_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE
get-type	Specifies the type of SNMP get operation to be applied to the object ID specified by the *oid-value* argument.
exact	Retrieves the object ID specified by the *oid-value* argument.
next	Retrieves the object ID that is the alphanumeric successor to the object ID specified by the *oid-value* argument.
entry-op	Compares the contents of the current object ID with the entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.

operator	Two-character string. The *operator* argument takes one of the following values: • **gt**—Greater than. • **ge**—Greater than or equal to. • **eq**—Equal to. • **ne**—Not equal to. • **lt**—Less than. • **le**—Less than or equal to.
entry-val	Specifies the value with which the contents of the current object ID are compared to decide if an SNMP event should be raised.
entry-value	Entry object ID value of the data element.
entry-type	Specifies a type of operation to be applied to the object ID specified by the *entry-value* argument.
value	Value is defined as the actual value of the *entry-value* or *exit-value* argument.
increment	Increment uses the *entry-value* or *exit-value* field as an incremental difference and the *entry-value* or *exit-value* is compared with the difference between the current counter value and the value when the event was last triggered (or the first polled sample if this is a new event). A negative value checks the incremental difference for a counter that is decreasing.
rate	Rate is defined as the average rate of change over a period of time. The time period is the *average-factor-value* multiplied by the *poll-int-value*. At each poll interval the difference between the current sample and the previous sample is taken and recorded as an absolute value. An average of the previous *average-factor-value* samples is taken to be the rate of change.
exit-comb	(Optional) Indicates the combination of exit conditions that must be met before event monitoring is reenabled.
or	(Optional) Specifies that an exit comparison operator and an exit object ID value or an exit time value must exist.
and	(Optional) Specifies that an exit comparison operator, an exit object ID value, and an exit time value must exist.
exit-op	(Optional) Compares the contents of the current object ID with the exit value using the specified operator. If there is a match, an event is triggered and event monitoring is reenabled.
exit-val	(Optional) Specifies the value with which the contents of the current object ID are compared to decide whether the exit criteria are met.
exit-value	(Optional) Exit object ID value of the data element.
exit-type	(Optional) Specifies a type of operation to be applied to the object ID specified by the *exit-value* argument. If not specified, the value is assumed.
exit-time	(Optional) Specifies the time period after which the event monitoring is reenabled. The timing starts after the event is triggered.
exit-time-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If only milliseconds are used, the format is 0.mmm.
exit-event	(Optional) Indicates whether a separate exit event is to be triggered when event monitoring is enabled after an initial event is triggered.

true	(Optional) Specifies that a separate exit event is triggered.
false	(Optional) Specifies that a separate exit event is not triggered. This is the default.
average-factor	(Optional) Specifies a number used to calculate the period used for rate-based calculations. The *average-factor-value* is multiplied by the *poll-int-value* to derive the period in milliseconds.
average-factor-value	(Optional) Number in the range from 1 to 64. The minimum average factor value is 1.
poll-interval	Specifies the time interval between consecutive polls.
poll-int-value	Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 1 to 4294967295. The range for milliseconds is from 0 to 999. The minimum polling interval is 1 second.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in SSSSSSSSSS[.MMM] format, where SSSSSSSSSS must be an integer representing seconds between 0 and 4294967295, inclusive, and where MMM must be an integer representing milliseconds between 0 and 999.

Command Default

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

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.3(14)T	Optional keywords to support SNMP rate-based events were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The **tag** and **maxrun** keywords and associated arguments were added.

Usage Guidelines

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

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

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

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

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

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

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

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

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

Examples

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

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

The following example shows an EEM applet called IPSLAping1 being registered to run when there is an exact match on the value of a specified SNMP object ID that represents a successful IP SLA ICMP echo operation (this is equivalent to a **ping** command). Four actions are triggered when the echo operation fails, and event monitoring is disabled until after the second failure.

A message saying that the ICMP echo operation to a server failed is sent to syslog, an SNMP trap is generated, EEM publishes an application-specific event, and a counter called IPSLA1F is incremented by a value of one.

```
Router(config)# event manager applet IPSLAping1
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.42.1.2.9.1.6.4 get-type exact
entry-op eq entry-val 1 exit-op eq exit-val 2 poll-interval 5
```

```
Router(config-applet)# action 1.0 syslog priority critical msg "Server IP echo failed:
OID=$_snmp_oid_val"
Router(config-applet)# action 1.1 snmp-trap strdata "EEM detected server reachability
failure to 10.1.88.9"
Router(config-applet)# action 1.2 publish-event sub-system 88000101 type 1 arg1 10.1.88.9
arg2 IPSLAEcho arg3 fail
```

Related Commands

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

event snmp-notification

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

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

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

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
oid	Specifies the SNMP object identifier (object ID) values in the *oid-val* argument as the event criteria.
oid-string	Object ID value of the data element, in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following: • COUNTER_TYPE • COUNTER_64_TYPE • GAUGE_TYPE • INTEGER_TYPE • OCTET_PRIM_TYPE • OPAQUE_PRIM_TYPE • TIME_TICKS_TYPE
oid-val *comparison-value*	Specifies the OID comparison value.
op	Compares the contents of the current object ID with the SNMP Protocol Data Unit (PDU) entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.
operator	Two-character string. The *operator* argument takes one of the following values: • **gt**—Greater than. • **ge**—Greater than or equal to. • **eq**—Equal to. • **ne**—Not equal to. • **lt**—Less than. • **le**—Less than or equal to.

maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999. The default value is 20 seconds.
src-ip-address	(Optional) Specifies the source IP address where the SNMP notification trap originates. The default is all; it is set to receive SNMP notification traps from all IP addresses.
ip-address	(Optional) The source IP address.
dest-ip-address	(Optional) Specifies the destination IP address where the SNMP notifications trap is sent. The default is all; it is set to receive SNMP traps from all destination IP addresses.
dest-ip-address	(Optional) The destination IP address.
default *seconds*	(Optional) Specifies the time period during which the snmp notification event detector waits for the policy to exit. The time period is specified in ssssssssss[.mmm] format, where ssssssssss must be an integer representing seconds between 0 and 4294967295 and mmm must be an integer representing milliseconds between 0 and 999.
direction	(Optional) Determines the direction of the SNMP trap or inform PDU to filter. The default is incoming. **incoming**—Specifies the incoming direction of the SNMP trap or inform PDU to filter. **outgoing**—Specifies the outgoing direction of the SNMP trap or inform PDU to filter.
msg-op	(Optional) Indicates the action to be taken on the SNMP PDU, drop it or send it once the event is triggered. **drop**—Specifies to drop the messages. **send**—Specifies to send the messages.

Command Default

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

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(20)T	This command was introduced.
15.0(1)M	This command was modified. The following keywords and arguments were added: **default**, *seconds*, **direction**, **incoming**, **outgoing**, **msg-op**, **drop**, and **send**.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

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

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

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

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

- INTEGER_TYPE
- COUNTER_TYPE
- GAUGE_TYPE
- TIME_TICKS_TYPE
- COUNTER_64_TYPE
- OCTET_PRIM_TYPE
- OPAQUE_PRIM_TYPE

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

The *operator* argument takes one of the following values:

- **gt**—Greater than.
- **ge**—Greater than or equal to.
- **eq**—Equal to.
- **ne**—Not equal to.
- **lt**—Less than.
- **le**—Less than or equal to.

Examples

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

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

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

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

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

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

Related Commands

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

event snmp-object

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

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

no event snmp-object

Syntax Description

oid	Specifies the SNMP object identifier (object ID).
oid-value	Object ID value of the data element in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value.
type *value*	Specifies the type of object. The following values are valid: • **counter**—A 32-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0. • **counter64**—A 64-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0. • **gauge**—A 32-bit number with a minimum value of 0. For example, the interface speed on a router is measured using a gauge object type. • **int**—A 32-bit number used to specify a numbered type within the context of a managed object. For example, to set the operational status of a router interface, 1 represents up and 2 represents down. • **ipv4**—IP version 4 address. • **octet**—An octet string in hex notation used to represent physical addresses. • **oid**—Object identifier value. • **string**—An octet string in text notation used to represent text strings. • **uint**—A 32-bit number used to represent decimal value.
sync	Specifies the SNMP and EEM policy execution. • **no**—Policy and SNMP will run asynchronously. • **yes**—Run policy and the result determines whether to run SNMP request.
skip	Mandatory if **sync** is set to **no** and should not be used if **sync** is **yes**. Specifies whether to skip CLI command execution. • **no**—CLI command should be executed. • **yes**—CLI command should not be executed.

istable	(Optional) Specifies whether the OID is a SNMP table. • **yes**—OID is an SNMP table. • **no**—IOD is not an SNMP table.
default	(Optional) The time period during which the SNMP Object event detector waits for the policy to exit.
seconds	(Optional) Number that represents seconds and optional milliseconds in the format ssssssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
maxrun	(Optional) Specifies the maximum runtime of the applet.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999. The default value is 20 seconds.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

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

Examples

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

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

Related Commands

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

event syslog

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run by matching syslog messages, use the **event syslog** command in applet configuration mode. To remove the syslog message event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **syslog pattern** *regular-expression* [**occurs** *num-occurrences*] [**period** *period-value*] [**priority** *priority-level*] [*severity-level*] [**maxrun** *maxruntime-number*]

no event [**tag** *event-tag*] **syslog pattern** *regular-expression* [**occurs** *num-occurrences*] [**period** *period-value*] [**priority** *priority-level*] [*severity-level*] [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
pattern	Specifies that a regular expression is used to perform the syslog message pattern match.
regular-expression	String value that is the pattern to be matched.
occurs	(Optional) Specifies the number of matching occurrences before an EEM event is triggered. If a number is not specified, an EEM event is triggered after the first match.
num-occurrences	(Optional) Integer in the range of 1 to 32, inclusive.
period	(Optional) Specifies the time interval during which the one or more occurrences must take place. If the **period** keyword is not specified, no time-period check is applied.
period-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
priority	(Optional) Specifies the priority level of the syslog messages to be screened. If this keyword is selected, the *priority-level* argument must be defined. If this keyword is not specified, the software will use the default of **priority all**, and all priorities will be considered when log messages are screened.
priority-level	(Optional) Number or name of the desired priority level against which syslog messages are matched. Messages at or numerically lower than the specified level are matched. Valid levels for the *priority-level* argument are as follows (enter the keyword or number, if available): • **all**—All priorities are considered when log messages are screened. • {**0** | **emergencies**}—System is unusable. • {**1** | **alerts**}—Immediate action is needed. • {**2** | **critical**}—Critical conditions. • {**3** | **errors**}—Error conditions.

	• {**4** | **warnings**}—Warning conditions. • {**5** | **notifications**}—Normal but significant conditions. • {**6** | **informational**}—Informational messages. • {**7** | **debugging**}—Debugging messages.
severity-level	(Optional) Specifies the severity level of the syslog messages to be screened. If no severity level is specified, the software will not use any severity filtering and all events will be considered when log messages are screened. The *severity-level* argument may be one or more of the following keywords: • **severity-critical**—Critical conditions. • **severity-debugging**—Debugging messages. • **severity-fatal**—Fatal conditions. • **severity-major**—Major conditions. • **severity-minor**—Minor conditions. • **severity-normal**—Normal conditions. • **severity-notification**—Significant conditions. • **severity-warning**—Warning conditions.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 4294967295, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM events are triggered on the basis of matches with syslog messages.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.3(14)T	Optional severity-level keywords were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.

Release	Modification
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The **tag** and **maxrun** keywords were added were added to support multiple event statements within an applet. The **entry-val-is-increment**, **true**, **false**, and **exit-val-is-increment**, keywords were deprecated.

Usage Guidelines

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

Valid levels for the *priority-level* argument are as follows (enter the keyword or number, if available):

- **all**—All priorities are considered when log messages are screened.
- {**0** | **emergencies**}—System is unusable.
- {**1** | **alerts**}—Immediate action is needed.
- {**2** | **critical**}—Critical conditions.
- {**3** | **errors**}—Error conditions.
- {**4** | **warnings**}—Warning conditions.
- {**5** | **notifications**}—Normal but significant conditions.
- {**6** | **informational**}—Informational messages.
- {**7** | **debugging**}—Debugging messages.

The *severity-level* argument may be one or more of the following keywords:

- **severity-critical**—Critical conditions.
- **severity-debugging**—Debugging messages.
- **severity-fatal**—Fatal conditions.
- **severity-major**—Major conditions.
- **severity-minor**—Minor conditions.
- **severity-normal**—Normal conditions.
- **severity-notification**—Significant conditions.
- **severity-warning**—Warning conditions

Examples

The following example shows how to specify an EEM applet to run when syslog identifies that Ethernet interface 1/0 is down. The applet sends a message about the interface to syslog.

```
Router(config)# event manager applet interface-down
Router(config-applet)# event syslog pattern {.*UPDOWN.*Ethernet1/0.*} occurs 4
```

Related Commands

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

event timer

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of time-specific events, use the **event timer** command in applet configuration mode. To remove the time-specific event criteria, use the **no** form of this command.

event [**tag** *event-tag*] **timer** {**absolute time** *time-value* | **countdown time** *time-value* | **cron cron-entry** *cron-entry* | **watchdog time** *time-value*} [**name** *timer-name*]

no event [**tag** *event-tag*] **timer** {**absolute time** *time-value* | **countdown time** *time-value* | **cron cron-entry** *cron-entry* | **watchdog time** *time-value*} [**name** *timer-name*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
absolute	Specifies that an event is triggered when the specified absolute time of day occurs.
time	Specifies the time interval during which the event must take place.
time-value	Integer that specifies, in seconds and optional milliseconds, the time interval during which the event must take place. The range for seconds is from 0 to 4294967295 and the range for milliseconds is from 0 to 999. The format is ssssss[.mmm]. When only milliseconds are specified, use the format 0.mmm.
countdown	Specifies that an event is triggered when the specified time counts down to zero. The timer does not reset.
cron	Specifies that an event is triggered when the CRON string specification matches the current time.
cron-entry	Specifies the first five fields of a UNIX crontab entry as used with the UNIX CRON daemon.
cron-entry	Text string that consists of five fields separated by spaces. The fields represent the times and dates when CRON timer events will be triggered. Fields and corresponding values are as follows: – *minute*—A number in the range from 0 to 59 that specifies when a CRON timer event is triggered. – *hour*—A number in the range from 0 to 23 that specifies when a CRON timer event is triggered. – *day-of-month*—A number in the range from 1 to 31 that specifies the day of the month when a CRON timer event is triggered. – *month*—A number in the range from 1 to 12 or the first three letters (not case-sensitive) of the name of the month in which a CRON timer event is triggered. – *day-of-week*—A number in the range from 0 to 6 (Sunday is 0) or the first three letters (not case-sensitive) of the name of the day when a CRON timer event is triggered. Instead of the first five fields, special strings can be entered. See the "Usage Guidelines" section for details.

watchdog	Specifies that an event is triggered when the specified time counts down to zero. The timer automatically resets to the initial value and continues to count down.
name	(Optional) Specifies that the timer is named.
timer-name	(Optional) Name of the timer.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the maxrun keyword is specified, the maxruntime-number value must be specified. If the maxrun keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 4294967295, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM events are triggered on the basis of time-specific events.

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.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(20)T	The **tag** and **maxrun** keywords were added were added to support multiple event statements within an applet. The **entry-val-is-increment**, **true**, **false**, and **exit-val-is-increment**, keywords were deprecated.

Usage Guidelines

For the *cron-entry* argument, the following special strings also are allowed in syntax:

- Range of numbers—The specified range is inclusive, and a hyphen separates the numbers. For example, 8-11 after the hour field specifies execution of a CRON timer event at hours 8, 9, 10, and 11.
- Asterisk (*)—Indicates that a field is not specified and can be any value.
- List—A list is a set of numbers or ranges separated by a comma but no space. For example, 1,2,5,9 or 0-4,8-12.

- Step value in conjunction with a range—Following a range with /*number* specifies skips of the *number* value through the range. For example, 0-23/2 in the hour field specifies that an event is triggered every second hour. Steps are permitted after an asterisk, for example */2 means every two hours.

Instead of the five fields of a UNIX crontab entry for the *cron-entry* argument, one of the following seven special strings can be entered:

- **@yearly**—An event is triggered once a year. This is the equivalent of specifying 0 0 1 1 * for the first five fields.
- **@annually**—Same as **@yearly**.
- **@monthly**—An event is triggered once a month. This is the equivalent of specifying 0 0 1 * * for the first five fields.
- **@weekly**—An event is triggered once a week. This is the equivalent of specifying 0 0 * * 0 for the first five fields.
- **@daily**—An event is triggered once a day. This is the equivalent of specifying 0 0 * * * for the first five fields.
- **@midnight**—Same as **@daily**.
- **@hourly**—An event is triggered once an hour. This is the equivalent of specifying 0 * * * * for the first five fields.

A CRON timer may not produce the intended result if the time-of-day clock is not set to the correct time. Network Time Protocol (NTP) services can be used to facilitate keeping an accurate time-of-day clock setting. For more details on NTP configuration, see the “Performing Basic System Management” chapter of the *Cisco IOS Network Management Configuration Guide*, Release 12.4.

Examples

The following example shows how to specify that an event is triggered one time after 5 hours:

```
Router(config)# event manager applet timer-absolute
Router(config-applet)# event timer absolute time 18000
```

The following example shows how to specify that an event is triggered once after 6 minutes and 6 milliseconds:

```
Router(config)# event manager applet timer-set
Router(config-applet)# event timer countdown time 360.006 name six-minutes
```

The following example shows how to specify that an event is triggered at 1:01 a.m. on January 1 each year:

```
Router(config)# event manager applet timer-cron1
Router(config-applet)# event timer cron cron-entry 1 1 1 1 * name Jan1
```

The following example shows how to specify that an event is triggered at noon on Monday through Friday of every week:

```
Router(config)# event manager applet timer-cron2
Router(config-applet)# event timer cron cron-entry 0 12 * * 1-5 name MonFri
```

The following example shows how to specify that an event is triggered at midnight on Sunday every week:

```
Router(config)# event manager applet timer-cron3
Router(config-applet)# event timer cron cron-entry @weekly name Sunday
```

The following example shows how to specify that an event is triggered every 5 hours:

```
Router(config)# event manager applet timer-watch
Router(config-applet)# event timer watchdog time 18000
```

Related Commands

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

event track

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

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

no event [*label*] [**tag** *event-tag*] **track** *object-number* [**state** {**up** | **down** | **any**}] [**maxrun** *maxruntime-number*]

Syntax Description

tag	(Optional) Specifies a tag using the *event-tag* argument that can be used with the **trigger** command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
label	(Optional) Unique identifier that can be any string. If the string contains embedded blanks, enclose it in double quotation marks.
object-number	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command.
state	(Optional) Specifies that the tracked object transition will cause an event to be raised.
up	(Optional) Specifies that an event will be raised when the tracked object transitions from a down state to an up state.
down	(Optional) Specifies that an event will be raised when the tracked object transitions from an up state to a down state.
any	(Optional) Specifies that an event will be raised when the tracked object transitions to or from any state. This is the default.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the **maxrun** keyword is specified, the *maxruntime-number* value must be specified. If the **maxrun** keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999).

Command Default

No EEM event criteria are specified.

Command Modes

Applet configuration (config-applet)

Command History

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

Release	Modification
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.4(20)T	The **tag** and **maxrun** keywords were added to support multiple event statements within an applet.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

There are two entry variables associated with this command:

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

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

Examples

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

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

Related Commands

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

event wdsysmon

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

event wdsysmon sub1 *subevent1* [**timewin** *timewin-value*] [**sub12-op** {**and** | **or** | **andnot**} **sub2** *subevent2*] [**node** *node-name*]

no event wdsysmon sub1 *subevent1* [**timewin** *timewin-value*] [**sub12-op** {**and** | **or** | **andnot**} **sub2** *subevent2*] [**node** *node-name*]

Subevent Syntax (for the *subevent1* and *subevent2* Arguments)

cpu-proc procname *process-name* **op** *operator* **val** *value* [**period** *period-value*]

cpu-tot op *operator* **val** *value* [**period** *period-value*]

deadlock procname *process-name*

dispatch-mgr procname *process-name* **op** *operator* **val** *value* [**period** *period-value*]

mem-proc procname *process-name* **op** *operator* **val** *value* [**is-percent** {**true** | **false**}] [**period** *period-value*]

mem-tot-avail op *operator* **val** *value* [**is-percent** {**true** | **false**}] [**period** *period-value*]

mem-tot-used op *operator* **val** *value* [**is-percent** {**true** | **false**}] [**period** *period-value*]

Syntax Description

sub1	Specifies the first subevent.
subevent1	First subevent. Use one of the seven forms of syntax shown above under the Subevent Syntax heading.
timewin	(Optional) Specifies the time window within which all the subevents must occur for an event to be generated.
timewin-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
sub12-op	(Optional) Indicates the combination operator for comparison between subevent 1 and subevent 2.
and	(Optional) Specifies that the results of both subevent 1 and subevent 2 must cross the specified thresholds.
or	(Optional) Specifies that the results of either subevent 1 or subevent 2 must cross the specified thresholds.
andnot	(Optional) Specifies that the results from subevent 1 must cross the specified threshold and the results from subevent 2 must not cross the specified threshold.
sub2	(Optional) Specifies the second subevent.

subevent2	(Optional) Second subevent. Use one of the seven forms of syntax shown above under the Subevent Syntax heading.
node	(Optional) Specifies the node.
node-name	(Optional) Node name.
Subevent Syntax	
cpu-proc	Specifies the use of a sample collection of CPU process statistics.
cpu-tot	Specifies the use of a sample collection of CPU total statistics.
deadlock	Specifies the use of a sample collection of deadlock statistics.
dispatch-mgr	Specifies the use of a sample collection of dispatch manager statistics.
mem-proc	Specifies the use of a sample collection of process memory statistics.
mem-tot-avail	Specifies the use of a sample collection of total available memory statistics.
mem-tot-used	Specifies the use of a sample collection of total used memory statistics.
procname	Specifies a Cisco IOS Software Modularity process name.
process-name	Name of the Software Modularity process to be monitored. If the process name contains embedded blanks, enclose it in double quotation marks.
op	Compares the collected CPU, deadlock, dispatch manager, or memory statistics sample with the value specified in the *value* argument. If there is a match, the subevent is triggered.
operator	Two-character string. The *operator* argument takes one of the following values: • **gt**—Greater than. • **ge**—Greater than or equal to. • **eq**—Equal to. • **ne**—Not equal to. • **lt**—Less than. • **le**—Less than or equal to.
val	Specifies the value with which the collected CPU, deadlock, dispatch manager, or memory statistics sample is compared to decide if the subevent should be raised.
value	Number in the range from 1 to 4294967295.
period	(Optional) Specifies the elapsed time period for the collection samples to be averaged.
period-value	(Optional) Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If only milliseconds are used, the format is 0.mmm. If the time period is 0, the most recent sample is used.
is-percent	(Optional) Indicates whether the *value* argument is a percentage.
true	(Optional) Specifies that the *value* argument is a percentage.
false	(Optional) Specifies that the *value* argument is not a percentage.

Command Default

No EEM events are triggered on the basis of Cisco IOS Software Modularity WDSysMon counters.

Command Modes

Applet configuration (config-applet)

Command History

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

Usage Guidelines

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

Examples

The following example shows how to configure a Cisco IOS Software Modularity policy to trigger an applet when the total amount of memory used by the process named "tcp.proc" has increased by more than 50 percent over the sample period of 60 seconds:

```
Router(config)# event manager applet WD_Sample
Router(config-applet)# event wdsysmon sub1 mem-proc procname "tcp.proc" op gt val 50
is-percent true period 60
Router(config-applet)# action 1 syslog msg "WD_Sample Policy Triggered"
```

Related Commands

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

exception core-file

To specify the name of the core dump file in Cisco IOS or Cisco IOS Software Modularity software, use the **exception core-file** command in global configuration mode. To return to the default core filename, use the **no** form of this command.

Cisco IOS Software

exception core-file *filename*

no exception core-file

Cisco IOS Software Modularity

exception core-file [*filename*] [**limit** *upper-limit*] [**compress**] [**timestamp**]

no exception core-file

Syntax Description

filename	Name of the core dump file saved on the server. (Optional) In Software Modularity images, if this argument is not specified, the default core file is named using the name of the process that is being dumped. For example, if the raw_ip.proc is the process that is being dumped, then the default core file is named raw_ip.proc.
limit	(Optional) For Cisco IOS Software Modularity images only. Specifies an upper limit of a range so that core dumps of more than one process can be created without overwriting the previous core dump.
upper-limit	(Optional) For Cisco IOS Software Modularity images only. Number, in the range from 1 to 64, that represents the upper limit.
compress	(Optional) For Cisco IOS Software Modularity images only. Turns on dump file compression. By default, compression is turned off.
timestamp	(Optional) For Cisco IOS Software Modularity images only. Adds a time stamp to the core dump file.

Command Default

Cisco IOS Software: The core file is named *hostname*-core, where *hostname* is the name of the router. Cisco IOS Software Modularity: The core file is named using the name of the process that is being dumped.

Command Modes

Global configuration (config)

Command History

Release	Modification
10.2	This command was introduced.
12.2(18)SXF4	The **limit**, **compress**, and **timestamp** keywords were added to support Software Modularity images.

Usage Guidelines

If you use TFTP to dump the core file to a server, the router will only dump the first 16 MB of the core file. If the router's memory is larger than 16 MB, the whole core file will not be copied to the server. Therefore, use rcp or FTP to dump the core file. The network dump is not supported in Software Modularity images.

Caution This command is of use only to Cisco technical support representatives in analyzing system failures in the field. Under normal circumstances, there should be no reason to change the default core filename. For that reason, this command should be used only by Cisco Certified Internetwork Experts (CCIEs) or under the direction of Cisco Technical Assistance Center (TAC) personnel.

Examples

Cisco IOS Software

In the following example, the router is configured to use FTP to dump a core file named dumpfile to the FTP server at 172.17.92.2 when the router crashes:

```
ip ftp username red
ip ftp password blue
exception protocol ftp
exception dump 172.17.92.2
exception core-file dumpfile
```

Cisco IOS Software Modularity

In the following example, the router is configured to dump the main memory used by the TCP process to a file named dump-tcp when the TCP process crashes. The dump file is configured with an upper limit of 20, to be compressed, and to have a time stamp applied.

```
exception core tcp.proc mainmem
exception core-file dump-tcp limit 20 compress timestamp
```

Note The **exception protocol** and **exception dump** commands are not supported in Software Modularity images.

Related Commands

Command	Description
exception core	Sets or changes the core dump options for a Cisco IOS Software Modularity process.
exception dump	Causes the router to dump a core file to a particular server when the router crashes.
exception memory	Causes the router to create a core dump and reboot when certain memory size parameters are violated.
exception protocol	Configures the protocol used for core dumps.
exception spurious-interrupt	Causes the router to create a core dump and reload after a specified number of spurious interrupts.
ip ftp password	Specifies the password to be used for FTP connections.
ip ftp username	Configures the username for FTP connections.

exception crashinfo buffersize

To change the size of the buffer used for crash info files, use the **exception crashinfo buffersize** command in global configuration mode. To revert to the default buffer size, use the **no** form of this command.

exception crashinfo buffersize *kilobytes*

no exception crashinfo buffersize *kilobytes*

Syntax Description

kilobytes	Buffer size, in kilobytes (KB). Range is 32 to 256. Default is 32.

Command Default

Crashinfo buffer is 32 KB.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(4)T, 12.2(11)	This command was introduced for the Cisco 3600 series only (3620, 3640, and 3660 platforms).
12.2(13)T	This command was implemented in 6400-NSP images.
12.4(6)T	This command was integrated into Cisco IOS Release 12.4(6)T.
12.2(18)SXF4	This command was integrated into Release 12.2(18)SXF4 to support Software Modularity images.

Usage Guidelines

The crash info file saves information that helps Cisco technical support representatives to debug problems that caused the Cisco IOS image to fail (crash). The device writes the crash information to the console at the time of the failure, and the file is created the next time you boot the Cisco IOS image after the failure (instead of while the system is failing).

Note If you are running a Software Modularity image, setting the crash info buffer size to the default of 32 KB does not limit the crash info buffer size. The crash info file size is limited to the value set if the value is set to anything other than the default 32 KB.

Examples

In the following example, the crash info buffer is set to 100 KB:

```
Router(config)# exception crashinfo buffersize 100
```

Related Commands

Command	Description
exception crashinfo file	Enables the creation of a diagnostic file at the time of unexpected system shutdowns.

exception crashinfo dump

To specify the type of output information to be written to the crashinfo file, use the **exception crashinfo dump** command in global configuration mode. To remove this information from the crashinfo file, use the **no** form of this command.

exception crashinfo dump {**command** *cli* | **garbage-detector**}

no exception crashinfo dump {**command** *cli* | **garbage-detector**}

Syntax Description

command *cli*	Indicates the Cisco IOS command for which you want the output information written to the crashinfo file.
garbage-detector	If a router crashes due to low memory, specifies that the output from the **show memory debug leaks summary** command should be written to the crashinfo file.

Command Default

This command is disabled by default.

If a router crashes due to low memory, the output from the following Cisco IOS commands is written to the crashinfo file by default:

- **show process memory**
- **show processes cpu**
- **show memory summary**
- **show buffers**

If the **exception crashinfo dump garbage-detector** command is enabled, the output from the **show memory debug leaks summary** command is also written to the crashinfo file by default.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

A benefit for using the **exception crashinfo dump** command is that it allows users to customize the crashinfo file to contain information that is relevant to their troubleshooting situation.

Examples

The following example shows how to specify that the output from the **show interfaces** command should be written to the crashinfo file:

```
exception crashinfo dump command show interfaces
```

Related Commands

Command	Description
exception memory	Sets free memory and memory block size threshold parameters.
show interfaces	Displays statistics for all interfaces configured on the router or access server.

exception crashinfo file

To enable the creation of a diagnostic file at the time of unexpected system shutdowns, use the **exception crashinfo file** command in global configuration mode. To disable the creation of crashinfo files, use the **no** form of this command.

exception crashinfo file *device*:*filename*

no exception crashinfo file *device*:*filename*

Syntax Description

device:filename	Specifies the flash device and file name to be used for storing the diagnostic information. The file name can be up to 38 characters. The colon is required.

Defaults

Enabled

Command Modes

Global configuration

Command History

Release	Modification
12.2(4)T, 12.2(11)	This command was introduced for the Cisco 3600 series only.
12.2(13)T	This command was implemented in 6400-NSP images.
12.4(6)T	This command was integrated into Cisco IOS Release 12.4(6)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The “crashinfo” file saves information that helps Cisco technical support representatives to debug problems that caused the Cisco IOS image to fail (crash). The switch writes the crash information to the console at the time of the failure, and the file is created the next time you boot the IOS image after the failure (instead of while the system is failing). The filename will be *filename_yyyymmdd-hhmmss*, where *y* is year, *m* is month, *d* is date, *h* is hour, and *s* is seconds.

Examples

In the following example, a crashinfo file called “crashdata” will be created in the default flash memory device if a system crash occurs:

```
Router(config)# exception crashinfo file flash:crashinfo
```

Related Commands

Command	Description
exception crashinfo buffersize	Changes the size of the crashinfo buffer.

exception crashinfo maximum files

To enable a Cisco IOS device to automatically delete old crashinfo files to help create space for the writing of new crashinfo files when a system crashes, use the **exception crashinfo maximum files** command in global configuration mode. To disable automatic deletion of crashinfo files, use the **no** form of this command.

exception crashinfo maximum files *file-numbers*

no exception crashinfo maximum files *file-numbers*

Syntax Description

file-numbers	The number of most recent crashinfo files across all file systems in the device to be saved when crashinfo files are deleted automatically. Valid values are from 0 to 32.

Command Default

This command is disabled by default.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced.
12.2(33)SRA	This feature was integrated in Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command is effective only when a device crashes. If the value of the *file-numbers* argument is given as zero (0), all old crashinfo files across all file systems are deleted when the crashinfo files are deleted automatically.

While booting a device, the default file location is bootflash.

If the file system does not have free space equivalent to or more than 250 KB, the system displays a warning. You can verify the available disk space and create free space for writing the crashinfo files.

Examples

The following example shows how to enable a Cisco IOS device to automatically delete old crashinfo files if the device needs space for writing new crashinfo files when a system crashes. In this example, the device is configured to preserve the 22 latest crashinfo files from previous crashinfo collections.

```
configure terminal
!
exception crashinfo maximum files 22
```

Related Commands

Command	Description
exception crashinfo buffersize	Changes the size of the crashinfo buffer.
exception crashinfo file	Creates a diagnostic file at the time of unexpected system shutdown.

exception data-corruption

To manage data error exceptions, use the **exception data-corruption** command in global configuration mode. To disable the management of data error exceptions, use the **no** form of this command.

exception data-corruption {**buffer** {**log** | **truncate**} | **reload**}

no exception data-corruption {**buffer** {**log** | **truncate**} | **reload**}

Syntax Description

buffer	Sets buffer corruption behavior.
log	Logs the number of attempts to overwrite the buffer.
truncate	Truncates the number of times the buffer is overwritten.
reload	Immediately reloads the data when a problem is detected.

Command Default

Data error exceptions are not managed.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.0(1)M	This command was introduced in a release earlier than Cisco IOS Release 15.0(1)M.
12.2(33)SXI	This command was integrated into a release earlier than Cisco IOS Release 12.2(33)SXI.

Examples

The following example shows how to enable the handling of data error exceptions:

```
Router(config)# exception data corruption buffer log
```

Related Commands

Command	Description
exception crashinfo	Facilitates the collection of crashinfo.

exception delay-dump

To pause or delay the dump of data error exceptions to the host, use the **exception delay-dump** command in global configuration mode. To disable the delay in the dump of data error exceptions to the host, use the **no** form of this command.

exception delay-dump *seconds*

no exception delay-dump

Syntax Description

seconds	Delay or pause time in seconds in the range 30 to 300. The default value is 30.

Command Default

The dump of data error exceptions is not delayed.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.0(1)M	This command was introduced in a release earlier than Cisco IOS Release 15.0(1)M.
12.2(33)SXI	This command was integrated into a release earlier than Cisco IOS Release 12.2(33)SXI.

Examples

The following example shows how to enable the handling of data error exceptions:

```
Router> enable
Router# configure terminal
Router(config)# exception delay-dump 32
```

Related Commands

Command	Description
exception crashinfo	Facilitates the collection of crashinfo.

exception dump

To configure the router to dump a core file to a particular server when the router crashes, use the **exception dump** command in global configuration mode. To disable core dumps, use the **no** form of this command.

exception dump *ip-address*

no exception dump

Syntax Description

ip-address	IP address of the server that stores the core dump file.

Defaults

Disabled

Command Modes

Global configuration

Command History

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

Usage Guidelines

Caution Use the **exception dump** command only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation. The resulting binary file, which is very large, must be transferred to a TFTP, FTP, or rcp server and subsequently interpreted by technical personnel that have access to source code and detailed memory maps.

If you use TFTP to dump the core file to a server, the router will only dump the first 16 MB of the core file. If the router's memory is larger than 16 MB, the whole core file will not be copied to the server. Therefore, use rcp or FTP to dump the core file.

The core dump is written to a file named *hostname*-core on your server, where *hostname* is the name of the router. You can change the name of the core file by configuring the **exception core-file** command.

This procedure can fail for certain types of system crashes. However, if successful, the core dump file will be the size of the memory available on the processor (for example, 16 MB for a CSC/4).

Examples

In the following example, a user configures a router to use FTP to dump a core file to the FTP server at 172.17.92.2 when it crashes:

```
Router(config)# ip ftp username red
Router(config)# ip ftp password blue
Router(config)# exception protocol ftp
Router(config)# exception dump 172.17.92.2
Router(config)# exception core-file dumpfile
```

Related Commands

Command	Description
exception core-file	Specifies the name of the core dump file.
exception memory	Causes the router to create a core dump and reboot when certain memory size parameters are violated.
exception protocol	Configures the protocol used for core dumps.
exception spurious-interrupt	Causes the router to create a core dump and reload after a specified number of spurious interrupts.
ip ftp password	Specifies the password to be used for FTP connections.
ip ftp username	Configures the username for FTP connections.
ip rcmd remote-username	Configures the remote username to be used when requesting a remote copy using rcp.

exception linecard

To enable storing of crash information for a line card and optionally specify the type and amount of information stored, use the **exception linecard** command in global configuration mode. To disable the storing of crash information for the line card, use the **no** form of this command.

exception linecard {**all** | **slot** *slot-number*} [**corefile** *filename* | **main-memory** *size* [**k** | **m**] | **queue-ram** *size* [**k** | **m**] | **rx-buffer** *size* [**k** | **m**] | **sqe-register-rx** | **sqe-register-tx** | **tx-buffer** *size* [**k** | **m**]]

no exception linecard

Syntax Description

all	Stores crash information for all line cards.
slot *slot-number*	Stores crash information for the line card in the specified slot. Slot numbers range from 0 to 11 for the Cisco 12012 and 0 to 7 for the Cisco 12008 router.
corefile *filename*	(Optional) Stores the crash information in the specified file in NVRAM. The default filename is *hostname*-**core**-*slot-number* (for example, c12012-core-8).
main-memory *size*	(Optional) Stores the crash information for the main memory on the line card and specifies the size of the crash information. Size of the memory to store is 0 to 268435456.
queue-ram *size*	(Optional) Stores the crash information for the queue RAM memory on the line card and specifies the size of the crash information. Size of the memory to store can be from 0 to 1048576.
rx-buffer *size* **tx-buffer** *size*	(Optional) Stores the crash information for the receive and transmit buffer on the line card and specifies the size of the crash information. Size of the memory to store can be from 0 to 67108864.
sqe-register-rx **sqe-register-tx**	(Optional) Stores crash information for the receive or transmit silicon queueing engine registers on the line card.
k **m**	(Optional) The **k** option multiplies the specified *size* by 1K (1024), and the **m** option multiplies the specified *size* by 1M (1024*1024).

Defaults

No crash information is stored for the line card.

If enabled with no options, the default is to store 256 MB of main memory.

Command Modes

Global configuration

Command History

Release	Modification
11.2 GS	This command was introduced for Cisco 12000 series Gigabit Switch Routers (GSRs).
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use caution when enabling the **exception linecard** global configuration command. Enabling all options could cause a large amount (150 to 250 MB) of crash information to be sent to the server.

Caution Use the **exception linecard** global configuration command only when directed by a technical support representative. Only enable options that the technical support representative requests you to enable. Technical support representatives need to be able to look at the crash information from the line card to troubleshoot serious problems on the line card. The crash information contains all the line card memory information including the main memory and transmit and receive buffer information. .

Examples

In the following example, the user enables the storing of crash information for line card 8. By default, 256 MB of main memory is stored.

```
Router(config)# exception linecard slot 8
```

exception memory

To set free memory and memory block size threshold parameters, use the **exception memory** command in global configuration mode. To disable this functionality, use the **no** form of this command.

exception memory {**fragment** | **minimum**} [**processor** | **io**] *size* [**interval 1**] [**reboot**] [**data overflow** {**fast** | **iomem** | **pcimem** | **processor** | **transient**}]

no exception memory {**fragment** | **minimum**} [**processor** | **io**] *size* [**interval 1**] [**reboot**] [**data overflow** {**fast** | **iomem** | **pcimem** | **processor** | **transient**}]

Syntax Description

fragment *size*	Sets the minimum contiguous block of memory in the free pool, in bytes.
minimum *size*	Sets the minimum size of the free memory pool, in bytes. The range is from 1 to 4090445040.
processor	(Optional) Specifies processor memory.
io	(Optional) Specifies I/O memory.
interval 1	(Optional) Checks the largest memory block size every 1 second. If the **interval 1** keyword is not configured, the memory block size is checked every 60 seconds (1 minute) by default.
reboot	(Optional) Reloads the router when a memory size threshold is violated. If the **reboot** keyword is not configured, the router will not reload when a memory size threshold is violated.
data overflow	(Optional) Enables data overflow detection for the following memory types: • **fast** • **iomem** • **pcimem** • **processor** • **transient**

Command Default

This command is disabled by default.

Command Modes

Global configuration (config)

Command History

Release	Modification
10.3	This command was introduced.
12.3(11)T	This command was modified. The **processor**, **io**, **interval 1**, and **reboot** keywords were added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was modified. The **data overflow** keyword was added.

Usage Guidelines

This command is used to troubleshoot memory leaks and memory fragmentation issues.

The free memory size is checked for every memory allocation. The largest memory block size is checked every 60 seconds by default. If the **interval 1** keyword is configured, the largest memory block size is checked every 1 second.

When a memory size threshold is violated, the router will display an error message and create a crashinfo file. A core dump file will also be created if the **exception dump** command is configured. The router will not reload unless the **reboot** keyword is configured.

Caution Use the **exception** commands only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation. The resulting binary file, which is very large, must be transferred to a TFTP, FTP, or rcp server and subsequently interpreted by technical personnel that have access to source code and detailed memory maps.

Examples

The following example shows how to configure the router to monitor the free memory. If the amount of free memory falls below 250,000 bytes, the router will create a crashinfo file and core dump file and reload.

```
configure terminal
!
exception dump 10.0.0.2
exception core-file memory.overrun
exception memory minimum 250000 reboot
```

Related Commands

Command	Description
exception core-file	Specifies the name of the core dump file.
exception crashinfo dump	Specifies the type of output information to be written to the crashinfo file.
exception dump	Configures the router to dump a core file to a particular server when the router crashes.
exception protocol	Configures the protocol used for core dumps.
exception region-size	Specifies the size of the region for the exception-time memory pool.
ip ftp password	Specifies the password to be used for FTP connections.
ip ftp username	Configures the username for FTP connections.

exception memory ignore overflow

To configure the Cisco IOS software to correct corruption in memory block headers and allow a router to continue its normal operation, use the **exception memory ignore overflow** command in global configuration mode. To disable memory overflow correction, use the **no** form of this command.

exception memory ignore overflow {**io** | **processor**} [**frequency** *seconds*] [**maxcount** *corrections*]

no exception memory ignore overflow {**io** | **processor**} [**frequency** *seconds*] [**maxcount** *corrections*]

Syntax Description

io	Selects input/output (also called packet) memory.
processor	Selects processor memory.
frequency *seconds*	(Optional) Specifies the minimum time gap between two memory block header corrections, in the range from 1 to 600 seconds. The default is once every 10 seconds.
maxcount *corrections*	(Optional) Specifies the maximum number of memory block header corrections allowed, in the range from 1 to 1000. The default is 0, which sets an unlimited number of corrections.

Command Default

The default is to allow the memory overflow correction once every 10 seconds, and for memory overflow corrections to happen an unlimited number of times.

Command Modes

Global configuration

Command History

Release	Modification
12.3(7)T	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use this command to improve device availability when software faults are detected in the network. You can configure the frequency and the maximum number of memory overflow corrections. If overflow correction is required more often than the configured value, a software forced reload is triggered because a severe system problem is indicated.

Examples

The following example shows how to set a maximum of five processor memory block header corruption corrections to occur every 30 seconds:

```
configure terminal
!
exception memory ignore overflow processor frequency 30 maxcount 5
end
```

Related Commands

Command	Description
show memory overflow	Displays the details of a memory block header corruption correction.

exception protocol

To configure the protocol used for core dumps, use the **exception protocol** command in global configuration mode. To configure the router to use the default protocol, use the **no** form of this command.

exception protocol {**ftp** | **rcp** | **tftp**}

no exception protocol

Syntax Description

ftp	Uses FTP for core dumps.
rcp	Uses rcp for core dumps.
tftp	Uses TFTP for core dumps. This is the default.

Defaults

TFTP

Command Modes

Global configuration

Command History

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

Usage Guidelines

Caution Use the **exception** commands only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation. The resulting binary file, which is very large, must be transferred to a TFTP, FTP, or rcp server and subsequently interpreted by technical personnel that have access to source code and detailed memory maps.

If you use TFTP to dump the core file to a server, the router will only dump the first 16 MB of the core file. If the router’s memory is larger than 16 MB, the whole core file will not be copied to the server. Therefore, use rcp or FTP to dump the core file.

Examples

In the following example, the user configures a router to use FTP to dump a core file to the FTP server at 172.17.92.2 when it crashes:

```
Router(config)# ip ftp username red
Router(config)# ip ftp password blue
Router(config)# exception protocol ftp
Router(config)# exception dump 172.17.92.2
```

Related Commands

Command	Description
exception core-file	Specifies the name of the core dump file.
exception dump	Causes the router to dump a core file to a particular server when the router crashes.
exception memory	Causes the router to create a core dump and reboot when certain memory size parameters are violated.
exception spurious-interrupt	Causes the router to create a core dump and reload after a specified number of spurious interrupts.
ip ftp password	Specifies the password to be used for FTP connections.
ip ftp username	Configures the username for FTP connections.

exception region-size

To specify the size of the region for the exception-time memory pool, use the **exception region-size** command in global configuration mode. To use the default region size, use the **no** form of this command.

exception region-size *size*

no exception region-size

Syntax Description

size	The size of the region for the exception-time memory pool.

Defaults

16,384 bytes

Command Modes

Global configuration

Command History

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

Usage Guidelines

Caution Use the **exception** commands only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation. The resulting binary file, which is very large, must be transferred to a TFTP, FTP, or rcp server and subsequently interpreted by technical personnel that have access to source code and detailed memory maps.

The **exception region-size** command is used to define a small amount of memory to serve as a fallback pool when the processor memory pool is marked corrupt. The **exception memory** command must be used to allocate memory to perform a core dump.

Examples

In the following example, the region size is set at 1024:

```
Router(config)# exception region-size 1024
```

Related Commands

Command	Description
exception core-file	Specifies the name of the core dump file.
exception dump	Configures the router to dump a core file to a particular server when the router crashes.
exception memory	Causes the router to create a core dump and reboot when certain memory size parameters are violated.
exception protocol	Configures the protocol used for core dumps.
ip ftp password	Specifies the password to be used for FTP connections.
ip ftp username	Configures the username for FTP connections.

exception spurious-interrupt

To configure the router to create a core dump and reload after a specified number of spurious interrupts, use the **exception spurious-interrupt** command in global configuration mode. To disable the core dump and reload, use the **no** form of this command.

exception spurious-interrupt [*number*]

no exception spurious-interrupt

Syntax Description

number	(Optional) A number from 1 to 4294967295 that indicates the maximum number of spurious interrupts to include in the core dump before reloading.

Defaults

Disabled

Command Modes

Global configuration

Command History

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

Usage Guidelines

Caution Use the **exception** commands only under the direction of a technical support representative. Creating a core dump while the router is functioning in a network can disrupt network operation. The resulting binary file, which is very large, must be transferred to a TFTP, FTP, or rcp server and subsequently interpreted by technical personnel that have access to source code and detailed memory maps.

If you use TFTP to dump the core dump file to a server, the router will only dump the first 16 MB of the file. If the router's memory is larger than 16 MB, the whole core file will not be copied to the server. Therefore, use rcp or FTP to dump the core file.

Examples

In the following example, the user configures a router to create a core dump with a limit of two spurious interrupts:

```
Router(config)# exception spurious-interrupt 2
```

Related Commands

Command	Description
exception core-file	Specifies the name of the core dump file.
ip ftp password	Specifies the password to be used for FTP connections.
ip ftp username	Configures the user name for FTP connections.

expression

To specify an expression for evaluation, use the **expression** command in expression configuration mode. To disable the configured settings, use the **no** form of this command.

expression *expression*

no expression

Syntax Description

expression	Expression to be evaluated.

Command Default

By default, no expression is configured.

Command Modes

Expression configuration (config-expression)

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

The expressions are in the American National Standards Institute for the C programming language (ANSI C) syntax except for the variable names. Variables are expressed as $ (dollar sign) and integers that correspond to the object number. An example of an expression is ($1-$5)*100.

Examples

The following example shows how to specify an expression:

```
Router(config)# snmp mib expression owner owner1 name expressionA
Router(config-expression)# expression ($1+$2)*800/$3
Router(config-expression)#
```

Related Commands

Command	Description
snmp mib expression owner	Specifies an expression.

falling (test_threshold)

To specify a falling threshold value for the threshold trigger test, use the **falling** command in event trigger threshold configuration mode. To disable the specified threshold, use the **no** form of this command.

falling {*threshold-value* | **event owner** *event-owner* **name** *event-name*}

no falling

Syntax Description

threshold-value	Numerical value for falling threshold. The default value is 0.
event	Specifies the event.
owner	Specifies the event owner.
event-owner	Name of the event owner.
name	Indicates the name of an event.
event-name	Name of an event.

Command Default

The default falling threshold value is 0. No event is invoked by default.

Command Modes

Event trigger threshold configuration (config-event-trigger-threshold)

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

The falling threshold value you specify is verified when the threshold trigger is active. If the sample value is equal to or less than the value you specify and greater than the value at the last sampling interval, a corresponding trigger is generated.

The **falling event owner** command specifies the event to invoke when the falling trigger fires. An event is identified by the owner and name, and is configured by using the **snmp mib event owner** command.

Examples

The following example shows how to specify a falling threshold value of 12:

```
Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test threshold
Router(config-event-trigger-threshold)# falling 12
Router(config-event-trigger-threshold)#
```

Related Commands

Command	Description
test	Enables a trigger test.

format (bulkstat)

To specify the format to be used for the bulk statistics data file, use the **format** command in Bulk Statistics Transfer configuration mode. To disable a previously configured format specification and return to the default, use the **no** form of this command.

format {**bulkBinary** | **bulkASCII** | **schemaASCII**}

no format

Syntax Description

bulkBinary	Binary format.
bulkASCII	ASCII (human-readable) format.
schemaASCII	ASCII format with additional bulk statistics schema tags. This is the default.

Command Default

The default bulk statistics transfer format is schemaASCII.

Command Modes

Bulk Statistics Transfer configuration (config-bulk-tr)

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS Release XE 2.1.

Usage Guidelines

Note In Cisco IOS Release 12.0(24)S, only the schemaASCII format is supported. This command will not change the file format in that release.

The bulk statistics data file (VFile) contains two types of fields: tags and data. Tags are used to set off data to distinguish fields of the file. All other information is in data fields.

For the bulkASCII and bulkBinary formats, periodic polling enables data for a single data group (object list) to be collected more than once in the same VFile. Each such instance of a data group can be treated as a different “table” type.

Every object and table tag contains an additional sysUpTime field. Similarly each row tag contains the value of the sysUpTime when the data for that row was collected. The sysUpTime provides a time stamp for the data.

For additional information about the structures of the bulk statistics data file formats, see the definitions in the CISCO-DATA-COLLECTION-MIB.

Examples

In the following example, the bulk statistics data file is set to schemaASCII:

```
Router(config)# snmp mib bulkstat transfer bulkstat1
Router(config-bulk-tr)# schema ATM2/0-IFMIB
Router(config-bulk-tr)# url primary ftp://user:pswrd@host/folder/bulkstat1
Router(config-bulk-tr)# format schemaASCII
Router(config-bulk-tr)# exit
```

Related Commands

Command	Description
snmp mib bulkstat transfer	Names a bulk statistics transfer configuration and enters Bulk Statistics Transfer configuration mode.

format global

To specify a default Operational Data Model (ODM) spec file other than the built-in spec file for XML-formatted requests, use the **format global** command in global configuration mode. To remove the default file, use the **no** form of this command.

format global *location*:*local-filename*

no format global

Syntax Description

location:*local-filename*	Command ODM file location and filename. Valid locations are **bootflash:**, **flash:**, **nvram:**, and any valid disk or slot number (such as **disk0:** or **slot1:**). ODM spec files have a .odm suffix.

Command Default

The built-in spec file is used.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was modified. It was integrated into Cisco IOS Release 12.2(33)SRE.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines

Use the **format global** command to specify an ODM spec file as the default for all XML-formatted requests coming from NETCONF operations. The NETCONF file search precedence is to look first for the file associated by the **netconf format** command, then for the file defined by the **format global** command, and finally for the built-in spec file.

The ODM spec file must exist on the filesystem before NETCONF can be configured to use it. If the file does not exist, the **format global** command is rejected.

Examples

The following example shows how to define a default ODM file to be used for all requests, then associates that file with NETCONF for all XML-formatted requests. If no file is specified, the built-in spec file is used for all requests:

```
Router(config)# format global disk0:spec3.3.odm
Router(config)# netconf format disk2:spec3.3.odm
```

Related Commands

Command	Description
netconf format	Associates NETCONF with an ODM spec file for XML-formatted requests.
spec-file install built-in	Replaces the current spec file with the built-in spec file.
spec-file install file	Replaces a local spec file with a remote spec file.

frequency

To specify an interval between trigger samples, use the **frequency** command in event trigger configuration mode. To disable the configured interval, use the **no** form of this command.

frequency *seconds*

no frequency

Syntax Description

seconds	Number of seconds between the two trigger samples. The default is 600.

Command Default

The interval between the trigger samples is set to the default value.

Command Modes

Event trigger configuration (config-event-trigger)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.
Cisco IOS XE Release 3.1S	This command was integrated into Cisco IOS XE Release 3.1S.

Usage Guidelines

The **frequency** command configures the waiting time between trigger samples. By default, the frequency of object sampling is 600 seconds.

Examples

The following example shows how to specify an interval of 360 seconds for sampling:

```
Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# frequency 360
Router(config-event-trigger)#
```

Related Commands

Command	Description
snmp mib event trigger owner	Specifies an event trigger owner while configuring an event trigger.

hw-module logging onboard (Cat 6K)

To re-enable onboard failure logging (OBFL) on Cisco Catalyst 6000 series switches if logging has been disabled, use the **hw-module logging onboard** command in global configuration mode. To disable OBFL (not recommended), use the **no** form of this command.

hw-module switch *switch-number* **module** *module-number* **logging onboard** [**message level** {**1-7**}]

no hw-module switch *switch-number* **module** *module-number* **logging onboard** [**message level** {**1-7**}]

Syntax Description

switch *switch-number*	Specifies the switch number.
module *module-number*	Specifies the module number.
message level {**1-7**}	(Optional) Specifies the level of severity for system messages that will be logged in OBFL files, as follows: Level 1—Alert (immediate action needed) Level 2—Critical condition Level 3—Error condition Level 4—Warning condition Level 5—Notification (significant condition) Level 6—Informational message only Level 7—Debugging (appears during debugging only)

Command Default

Enabled in all hardware and is the recommended state; all levels of system messages are logged.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(33)SXH	This command was introduced.

Usage Guidelines

This command enables operating temperatures, hardware uptime, interrupts, and other important events and messages to be recorded in files stored in nonvolatile memory, so that the data can be used to diagnose problems with hardware cards installed in a Cisco router or switch. When the onboard hardware is started up, a first record is made for each area monitored and becomes a base value for subsequent records. This command provides a circular updating scheme for collecting continuous records and archiving older (historical) records, ensuring accurate data about the hardware. Data is recorded in one of two formats: continuous information that displays a snapshot of data in a continuous file, and summary information that provides details about the data being collected. Use the **show logging onboard** privileged EXEC command to see reports of current and historical data.

This configuration command is applicable to the module inserted in a device. When the module is removed and inserted into a new device, the configuration of this command follows the module to the new device.

This command is normally accessed through the route processor or supervisor command line interface; however, some system images do not provide full support for client remote terminal access. When using these images, use the **attach** command to connect to the console on the line card.

Examples

The following example shows how to configure OBFL message logging at level 7 (debugging):

```
Router> enable
Router# configure terminal
Router(config)# hw-module switch 2 module 1 logging onboard message level 7
Router(config)# end
```

Related Commands

Command	Description
attach	Connects to a specific line card for the purpose of executing commands on that card.
clear logging onboard (Cat 6K)	Clears onboard failure logs.
copy logging onboard (Cat 6K)	Copies OBFL data from the target OBFL-enabled module to a local or remote file system.
show logging onboard (Cat 6K)	Displays onboard failure logs.