Embedded Event Manager 2.0

Embedded Event Manager (EEM) is a distributed, scalable, and customized approach to event detection and recovery offered directly in a Cisco IOS device.

Feature History for Embedded Event Manager 2.0

Release	Modification
12.2(25)S	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Embedded Event Manager 2.0

- If the **action cns-event** command is used, access to a CNS Event gateway must be configured.
- If the **action force-switchover** command is used, a secondary processor must be configured on the device.
- If the **action snmp-trap** command is used, the **snmp-server enable traps event-manager** command must be enabled to permit Simple Network Management Protocol (SNMP) traps to be sent from the Cisco IOS device to the SNMP server. Other relevant **snmp-server** commands must also be configured; for details see action snmp-trap command page.

Restrictions for Embedded Event Manager 2.0

Only Cisco-defined policies can be used with Cisco IOS Release 12.2(25)S.

Information About Embedded Event Manager 2.0

To configure Embedded Event Manager 2.0, you should understand the following concepts:

- Embedded Event Manager 2.0, page 2
- Event Detectors, page 3
- Embedded Event Manager Actions, page 5
- Embedded Event Manager Environment Variables, page 5

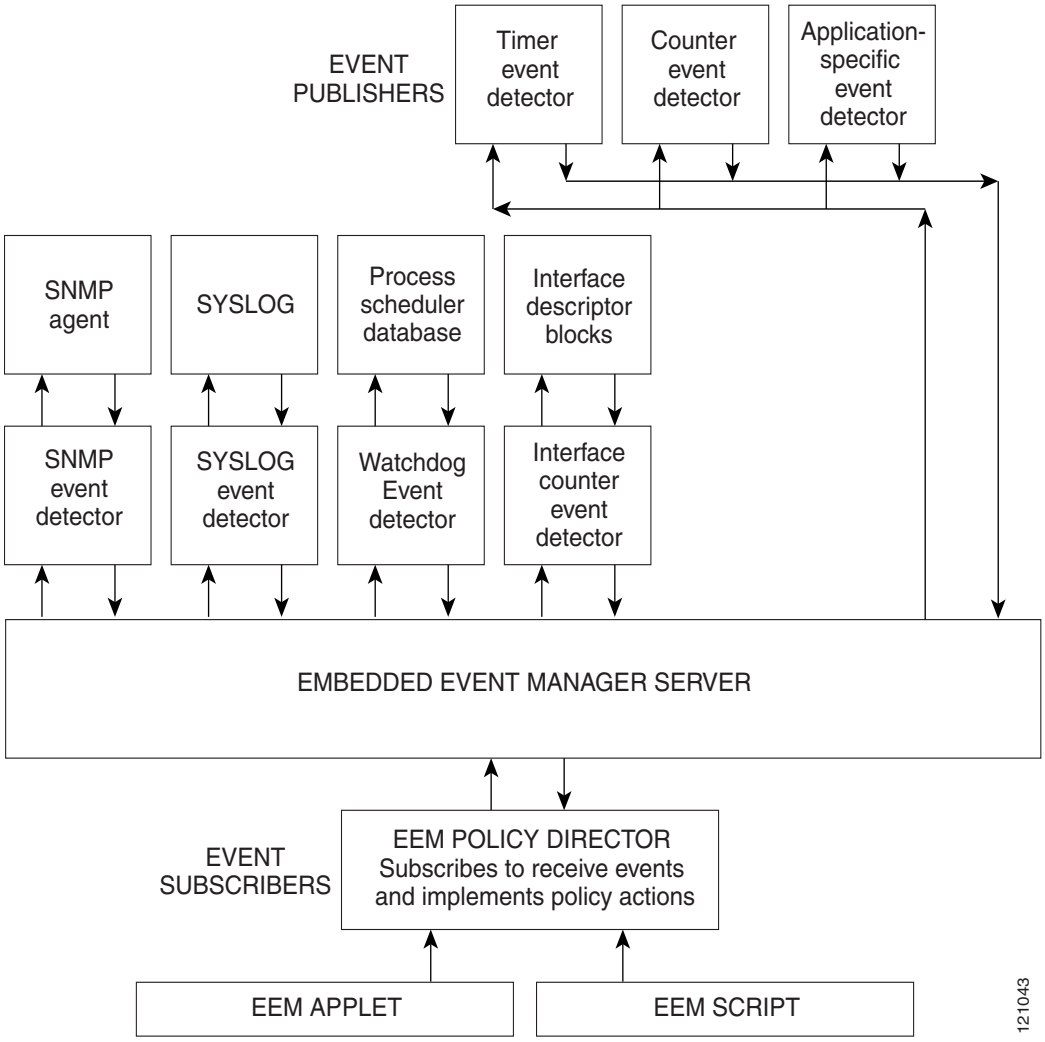
Embedded Event Manager 2.0

Event tracking and management has traditionally been performed by devices external to the networking device. Embedded Event Manager (EEM) has been designed to offer event management capability directly in Cisco IOS based devices. The on-device, proactive event management capabilities of EEM are useful because not all event management can be done off router because some problems compromise communication between the router and the external network management device. Capturing the state of the router during such situations can be invaluable in taking immediate recovery actions and gathering information to perform root-cause analysis. Network availability is also improved if automatic recovery actions are performed without the need to fully reboot the routing device.

EEM 2.0 is a flexible, policy-driven framework that supports in-box monitoring of different components of the system with the help of software agents known as event detectors. Figure 1 shows the relationship between the EEM server, the event publishers (event detectors), and the event subscribers (policies). Basically, event publishers screen events and publish them when there is a match on an event specification that is provided by the event subscriber. Event detectors notify the EEM when an event of interest occurs. The EEM policies that are configured using the Cisco IOS command-line interface (CLI) then implement recovery on the basis of the current state of the system and the actions specified in the policy for the given event.

EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs. There are two types of EEM policies: an applet or a script. An applet is a simple form of policy that is defined within the CLI configuration. A script is a form of policy that is written in Tool Command Language (Tcl).

Figure 1 Embedded Event Manager 2.0



Event Detectors

Events are detected by software routines called event detectors. Event detectors are separate systems that provide an interface between the agent being monitored, for example Simple Network Management Protocol (SNMP), and the EEM policies where an action can be implemented. EEM 2.0 contains the following event detectors.

Application-Specific Event Detector

The application-specific event detector allows any Embedded Event Manager policy to publish an event.

Counter Event Detector

The counter event detector publishes an event when a named counter crosses a specified threshold. There are two or more participants that affect counter processing. The counter event detector can modify the counter, and one or more subscribers define the criteria that cause the event to be published. After a counter event has been published, the counter monitoring logic can be reset to start monitoring the counter immediately or it can be reset when a second threshold—called an exit value—is crossed.

Interface Counter Event Detector

The interface counter event detector publishes an event when a generic Cisco IOS interface counter for a specified interface crosses a defined threshold. A threshold can be specified as an absolute value or an incremental value. If the incremental value is set to 50, for example, an event would be published when the interface counter increases by 50.

After an interface counter event has been published, the interface counter monitoring logic is reset using two methods. The interface counter is reset either when a second threshold—called an exit value—is crossed or when an elapsed period of time occurs.

SNMP Event Detector

The SNMP event detector allows a standard SNMP MIB object to be monitored and an event to be generated when the object matches specified values or crosses specified thresholds.

Syslog Event Detector

The syslog event detector allows for screening syslog messages for a regular expression pattern match. The selected messages can be further qualified, requiring that a specific number of occurrences be logged within a specified time. A match on a specified event criteria triggers a configured policy action.

Timer Event Detector

The timer event detector publishes events for the following four different types of timers:

- An absolute-time-of-day timer publishes an event when a specified absolute date and time occurs.
- A countdown timer publishes an event when a timer counts down to zero.
- A watchdog timer publishes an event when a timer counts down to zero and then the timer automatically resets itself to its initial value and starts to count down again.
- A CRON timer publishes an event using a UNIX standard CRON specification to indicate when the event is to be published. A CRON timer never publishes events more than once per minute.

Watchdog Event Detector

The Cisco IOS watchdog event detector publishes an event when one of the following occurs:

- CPU utilization for a Cisco IOS process crosses a threshold.
- Memory utilization for a Cisco IOS process crosses a threshold.

Two events may be monitored at the same time, and the event publishing criteria can be specified to require one event or both events to cross their specified thresholds.

Embedded Event Manager Actions

The CLI-based corrective actions that are taken when event detectors report events enable a powerful on-device event management mechanism. EEM 2.0 supports the following actions:

- Modifying a named counter.
- Publishing an application-specific event.
- Generating an SNMP trap.
- Generating prioritized syslog messages.
- Generating a CNS event for upstream processing by Cisco CNS devices.
- Reloading the Cisco IOS software.
- Switching to a secondary processor in a fully redundant hardware configuration.

Embedded Event Manager Environment Variables

Tool Command Language (Tcl) permits global variables—called environment variables—to be defined for use within an EEM policy. There are three different types of environment variables associated with Embedded Event Manager. User-defined environment variables are defined by you if you create an environment variable in a policy that you have written. Cisco-defined environment variables are either created for a specific policy (sample policy) or considered to be Cisco system-defined (see Table 1) and may apply to one specific event detector or to all event detectors. Environment variables that are user-defined or defined by Cisco in a sample policy are set using the **event manager environment** command. Variables that are used in the EEM policy must be defined before you register the policy. A policy contains a section called "Environment Must Define" that can be defined to check that any required environment variables are defined before the policy runs. Cisco system-defined environment variables are set by the system when the policy starts to execute.

Note Cisco-defined environment variables begin with an underscore character (_). We strongly recommend that customers avoid the same naming convention to prevent naming conflicts.

Table 1 describes the Cisco system-defined environment variables.

Table 1 Cisco System-Defined Environment Variables

Environment Variable	Description
All Events	
_event_pub_time	Time at which the event type was published.
_event_type_string	Event type that triggered the event.
Application-Specific Event Detector	
_application_data1	Character text, the value of an environment variable, or a combination of the two to be passed to an application-specific event when the event is published.
_application_data2	Character text, the value of an environment variable, or a combination of the two to be passed to an application-specific event when the event is published.

Table 1 Cisco System-Defined Environment Variables (continued)

Environment Variable	Description
_application_data3	Character text, the value of an environment variable, or a combination of the two to be passed to an application-specific event when the event is published.
_application_data4	Character text, the value of an environment variable, or a combination of the two to be passed to an application-specific event when the event is published.
_application_sub_system	Event application subsystem number.
_application_type	Type of application.
Counter Event Detector	
_counter_name	Name of counter.
_counter_value	Value of counter.
Interface Counter Event Detector	
_interface_is_increment	Text False or True to indicate whether the current interface counter value is an absolute or an increment value.
_interface_name	Name of the interface to be monitored.
_interface_parameter	Name of the interface counter to be monitored.
_interface_value	Value with which the current interface counter value is compared.
SNMP Event Detector	
_snmp_oid	The SNMP object ID that caused the event to be published.
_snmp_oid_value	The SNMP object ID value when the event was published.
Syslog Event Detector	
_syslog_msg	The syslog message that caused the event to be published.
Timer Event Detector	
_timer_remain	Time available before the timer expires. **Note** This environment variable is not available for the CRON timer.
_timer_time	Time at which the last event was triggered.
_timer_type	Type of timer.
Watchdog Event Detector	
_ioswd_node	Slot number for the route processor (RP) reporting node.
_ioswd_num_subs	Number of subevents present.
All Watchdog Subevents	
_ioswd_sub1_present _ioswd_sub2_present	Indicates if subevent 1 or subevent 2 is present. A value of 1 means that the subevent is present; a value of 0 means that the subevent is not present.
_ioswd_sub1_type _ioswd_sub2_type	Event type, either cpu_util or mem_used.
Watchdog cpu_util Events	

Table 1 Cisco System-Defined Environment Variables (continued)

Environment Variable	Description
_ioswd_sub1_path _ioswd_sub2_path	Process name of subevents.
_ioswd_sub1_period _ioswd_sub2_period	Time period, in seconds and optional milliseconds, used for measurement in subevents.
_ioswd_sub1_pid _ioswd_sub2_pid	Process identifier of subevents.
_ioswd_sub1_taskname _ioswd_sub2_taskname	Task name of subevents.
_ioswd_sub1_value _ioswd_sub2_value	CPU utilization of subevents measured as a percentage.
Watchdog mem_used Events	
_ioswd_sub1_diff _ioswd_sub2_diff	Percentage value of the difference that triggered the event. **Note** This variable is set only when the _ioswd_subx_is_percent variable contains a value of 1.
_ioswd_sub1_is_percent _ioswd_sub2_is_percent	Identifies whether the value is a percentage. A value of 0 means that the value is not a percentage; a value of 1 means that the value is a percentage.
_ioswd_sub1_path _ioswd_sub2_path	Process name of subevents.
_ioswd_sub1_pid _ioswd_sub2_pid	Process identifier of subevents.
_ioswd_sub1_taskname _ioswd_sub2_taskname	Task name of subevents.
_ioswd_sub1_value _ioswd_sub2_value	CPU utilization of subevents measured as a percentage.

How to Configure Embedded Event Manager 2.0

This section contains the following tasks:

- Registering and Defining an Embedded Event Manager Applet, page 7 (required)
- Registering and Defining an Embedded Event Manager Tcl Script, page 9 (required)
- Displaying Embedded Event Manager History Data, page 11 (optional)
- Displaying Embedded Event Manager Registered Policies, page 12 (optional)

Registering and Defining an Embedded Event Manager Applet

Perform this task to register an applet with Embedded Event Manager and to define the EEM applet using event applet and action applet commands. Only one event applet command is allowed in an EEM applet. Multiple action applet commands are permitted. If no event and no action commands are specified, the applet is removed when you exit configuration mode.

EEM Policies

EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs. There are two types of EEM policies: an applet or a script. An applet is a simple form of policy that is defined within the CLI configuration. A script is a form of policy that is written in Tcl.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **event manager applet** *applet-name*
4. **event snmp oid** *oid-value* **get-type** {**exact** | **next**} **entry-op** *operator* **entry-val** *entry-value* [**exit-comb** {**or** | **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-time** *exit-time-value*] **poll-interval** *poll-int-value*
5. **action** *label* **syslog** [**priority** *priority-level*] **msg** *msg-text*
6. Repeat Step 5.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**event manager applet** *applet-name* **Example:** `Router(config)# event manager applet memory-fail`	Registers the applet with the Embedded Event Manager (EEM) and enters applet configuration mode.
Step 4	**event snmp oid** *oid-value* **get-type** {**exact** | **next**} **entry-op** *operator* **entry-val** *entry-value* [**exit-comb** {**or** | **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-time** *exit-time-value*] **poll-interval** *poll-int-value* **Example:** `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`	Specifies the event criteria that cause the EEM applet to run. • In this example, an EEM event is triggered when one of the fields specified by an SNMP object ID crosses a defined threshold. • Exit criteria are optional, and if not specified, event monitoring is reenabled immediately.

	Command or Action	Purpose
Step 5	**action** *label* **syslog** [**priority** *priority-level*] **msg** *msg-text* **Example:** `Router(config-applet)# action 1.0 syslog priority critical msg "Memory exhausted; current available memory is $_snmp_oid_val bytes"`	Specifies the action to be taken when an EEM applet is triggered. • In this example, the action to be taken is to write a message to syslog. • The optional **priority** keyword specifies the priority level of the syslog messages. If selected, the *priority-level* argument must be defined. • The *msg-text* argument can be character text, an environment variable, or a combination of the two.
Step 6	Repeat Step 5. **Example:** `Router(config-applet)# action 2.0 force-switchover`	(Optional) Repeat Step 5 to add other action CLI commands to the applet.
Step 7	**end** **Example:** `Router(config-applet)# end`	Exits applet configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug event manager** command in privileged EXEC mode to troubleshoot EEM command operations. Use any debugging command with caution as the volume of generated output can slow or stop the router operations. We recommend that this command be used only under the supervision of a Cisco engineer.

Registering and Defining an Embedded Event Manager Tcl Script

Perform this task to configure environment variables and register an EEM policy. EEM schedules and runs policies on the basis of an event specification that is contained within the policy itself. When an EEM policy is registered, the software examines the policy and registers it to be run when the specified event occurs.

EEM Policies

EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs. There are two types of EEM policies: an applet or a script. An applet is a simple form of policy that is defined within the CLI configuration. A script is a form of policy that is written in Tcl.

Prerequisites

You must have a policy available that is written in the Tcl scripting language. A sample policy is provided that can be stored in the system policy directory.

SUMMARY STEPS

1. **enable**
2. **show event manager environment** [**all** | *variable-name*]
3. **configure terminal**
4. **event manager environment** *variable-name string*
5. Repeat Step 4 for all the required environment variables.
6. **event manager policy** *policy-file-name* [**type system**] [**trap**]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show event manager environment** [**all**|*variable-name*] **Example:** `Router# show event manager environment all`	(Optional) Displays the name and value of EEM environment variables. • The optional **all** keyword displays all the EEM environment variables. • The optional *variable-name* argument displays information about the specified environment variable.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**event manager environment** *variable-name string* **Example:** `Router(config)# event manager environment _cron_entry 0-59/2 0-23/1 * * 0-6`	Configures the value of the specified EEM environment variable. • In this example, the software assigns a CRON timer environment variable to be set to every second minute, every hour of every day.
Step 5	Repeat Step 4 for all the required environment variables.	Repeat Step 4 to configure all the environment variables required by the policy to be registered in Step 6.
Step 6	**event manager policy** *policy-file-name* [**type system**] [**trap**] **Example:** `Router(config)# event manager policy tm_countdown_ios.tcl type system`	Registers the EEM policy to be run when the specified event defined within the policy occurs. • In this example, the sample EEM policy named tm_countdown_ios.tcl is registered as a system policy.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Examples

Sample Output for the show event manager environment Command

In the following example, the **show event manager environment** privileged EXEC command is used to display the name and value of all EEM environment variables.

```
Router# show event manager environment all

No.  Name                          Value
1    _cron_entry                   0-59/2 0-23/1 * * 0-6
2    _show_cmd                     show ver
3    _syslog_pattern               .*UPDOWN.*Ethernet1/0.*
4    _config_cmd1                  interface Ethernet1/0
5    _config_cmd2                  no shut
```

Displaying Embedded Event Manager History Data

Perform this optional task to change the size of the history tables and to display EEM history data.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **event manager history size** {**events** | **traps**} [*size*]
4. **exit**
5. **show event manager history events** [**detailed**] [**maximum** *number*]
6. **show event manager history traps** {**server** | **policy**}

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **configure terminal**

Enters global configuration mode.

```
Router# configure terminal
```

Step 3 **event manager history size** {**events** | **traps**} [*size*]

Use this command to change the size of the EEM event history table or the size of the EEM SNMP trap history table. In the following example, the size of the EEM event history table is changed to 30 entries:

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

Step 4 **exit**

Exits global configuration mode and returns to privileged EXEC mode.

```
Router(config)# exit
```

Step 5 **show event manager history events** [**detailed**] [**maximum** *number*]

Use this command to display detailed information about each EEM event, for example:

```
Router# show event manager history events

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

Step 6 **show event manager history traps** {**server** | **policy**}

Use this command to display the EEM SNMP traps that have been sent either from the EEM server or from an EEM policy. In the following example, the EEM SNMP traps that were triggered from within an EEM policy are displayed.

```
Router# show event manager history traps policy

No.  Time                      Trap Type           Name
1    Wed Aug18  22:30:58 2004  policy              EEM Policy Director
2    Wed Aug18  22:34:58 2004  policy              EEM Policy Director
3    Wed Aug18  22:51:18 2004  policy              EEM Policy Director
```

Displaying Embedded Event Manager Registered Policies

Perform this optional task to display EEM registered policies.

SUMMARY STEPS

1. **enable**
2. **show event manager policy registered** [**event-type** *event-name*] [**time-ordered** | **name-ordered**]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show event manager policy registered** [**event-type** *event-name*] [**time-ordered** | **name-ordered**]

Use this command with the **time-ordered** keyword to display information about currently registered policies sorted by time, for example:

```
Router# show event manager policy registered time-ordered

No.  Type    Event Type          Time                    Registered Name
1    applet  snmp                Thu May30 05:57:16 2004 memory-fail
 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
```

```
 action 1.0 syslog priority critical msg Memory exhausted; current available memory is
$_snmp_oid_val bytes
 action 2.0 force-switchover
2    applet  syslog              Wed Jul16 00:05:17 2004 intf-down
 pattern {.*UPDOWN.*Ethernet1/0.*}
 action 1.0 cns-event msg Interface state change: $_syslog_msg
```

Use this command with the **name-ordered** keyword to display information about currently registered policies sorted by name, for example:

```
Router# show event manager policy registered name-ordered

No.  Type    Event Type          Time Registered          Name
1    applet  syslog              Wed Jul16  00:05:17 2004 intf-down
 pattern {.*UPDOWN.*Ethernet1/0.*}
 action 1.0 cns-event msg Interface state change: $_syslog_msg
2    applet  snmp                Thu May30 05:57:16 2004   memory-fail
 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
 action 1.0 syslog priority critical msg Memory exhausted; current available memory is
$_snmp_oid_val bytes
 action 2.0 force-switchover
```

Use this command with the **event-type** keyword to display information about currently registered policies for the event type specified in the *event-name* argument, for example:

```
Router# show event manager policy registered event-type syslog

No.  Type    Event Type          Time Registered          Name
1    applet  syslog              Wed Jul16  00:05:17 2004 intf-down
 pattern {.*UPDOWN.*Ethernet1/0.*}
 action 1.0 cns-event msg Interface state change: $_syslog_msg
```

Configuration Examples for Embedded Event Manager 2.0

This section contains the following configuration examples:

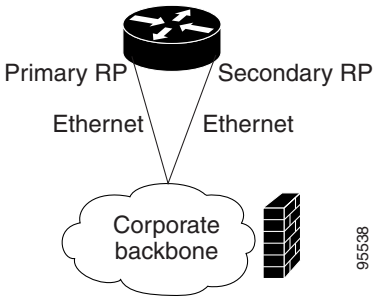
Embedded Event Manager Applet Configuration: Example

The following example shows how to configure an EEM applet that causes a switch to the secondary (redundant) Route Processor (RP) when the primary RP runs low on memory.

This example illustrates a method for taking preventative action against a software fault that causes a memory leak. The action taken here is designed to reduce downtime by switching over to a redundant RP when a possible memory leak is detected.

Figure 2 shows a dual RP router that is running an EEM image. An EEM applet has been registered through the CLI using the **event manager applet** command. The applet will run when the available memory on the primary RP falls below the specified threshold of 5,120,000 bytes. The applet actions are to write a message to syslog that indicates the number of bytes of memory available and to switch to the secondary RP.

Figure 2 Dual RP Topology



The commands used to register the policy are shown below.

```
event manager applet memory-demo
 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
 action 1.0 syslog priority critical msg "Memory exhausted; current available memory is
$_snmp_oid_val bytes"
 action 2.0 force-switchover
```

The registered applet is displayed using the **show event manager policy registered** command:

```
Router# show event manager policy registered

No.  Type    Event Type          Time Registered           Name
1    applet  snmp                Thu Jan30  05:57:16 2003  memory-demo
 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
 action 1.0 syslog priority critical msg Memory exhausted; current available memory is
$_snmp_oid_val bytes
 action 2.0 force-switchover
```

For the purpose of this example, a memory depletion is forced on the router, and a series of **show memory** commands are executed to watch the memory deplete:

```
Router# show memory

                Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   53585260   212348444   119523060    92825384    92825384    92365916
Fast        53565260      131080       70360       60720       60720       60668

Router# show memory

                Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   53585260   212364664   164509492    47855172    47855172    47169340
Fast        53565260      131080       70360       60720       60720       60668

Router# show memory

                Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   53585260   212369492   179488300    32881192    32881192    32127556
Fast        53565260      131080       70360       60720       60720       60668
```

When the threshold is reached, an EEM event is triggered. The applet named memory-demo runs, causing a syslog message to be written to the console and a switch to be made to the secondary RP. The following messages are logged:

```
00:08:31: %HA_EM-2-LOG: memory-demo: Memory exhausted; current available memory is 4484196
bytes
00:08:31: %HA_EM-6-FMS_SWITCH_HARDWARE: fh_io_msg: Policy has requested a hardware
switchover
```

Configuration for the Primary RP and Secondary RP

The following is partial output from the **show running-config** command on both the primary RP and the secondary (redundant) RP:

```
redundancy
 mode sso
.
.
.
!
event manager applet memory-demo
 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
 action 1.0 syslog priority critical msg "Memory exhausted; current available memory is
$_snmp_oid_val bytes"
 action 2.0 force-switchover
```

Embedded Event Manager Watchdog Event Detector Configuration: Example

The following example shows how to configure three EEM applets to demonstrate how the watchdog event detector works.

Watchdog Sample1 Policy

The first policy triggers an applet when the average CPU usage for the process named “IP Input” is greater than or equal to 1 percent for 10 seconds:

```
event manager applet IOSWD_Sample1
 event ioswdsysmon sub1 cpu-proc taskname “IP Input” op ge val 1 period 10
 action 1 syslog msg “IOSWD_Sample1 Policy Triggered”
```

Watchdog Sample2 Policy

The second policy triggers an applet when the total amount of memory used by the process named “Net Input” is greater than 100 kb:

```
event manager applet IOSWD_Sample2
 event ioswdsysmon sub1 mem-proc taskname “Net Input” op gt val 100 is-percent false
 action 1 syslog msg “IOSWD_Sample2 Policy Triggered”
```

Watchdog Sample3 Policy

The third policy triggers an applet when the total amount of memory used by the process named “IP RIB Update” has increased by more than 50 percent over the sample period of 60 seconds:

```
event manager applet IOSWD_Sample3
 event ioswdsysmon sub1 mem-proc taskname “IP RIB Update” op gt val 50 is-percent true
period 60
 action 1 syslog msg “IOSWD_Sample3 Policy Triggered”
```

The three policies are configured, and then repetitive large pings are made to the networking device from several workstations, causing the networking device to register some usage. This will trigger policies 1 and 2, and the console will display the following messages:

```
00:42:23: %HA_EM-6-LOG: IOSWD_Sample1: IOSWD_Sample1 Policy Triggered
00:42:47: %HA_EM-6-LOG: IOSWD_Sample2: IOSWD_Sample2 Policy Triggered
```

To view the policies that are registered, use the **show event manager policy registered** command:

```
Router# show event manager policy registered

No.  Class   Type    Event Type          Trap  Time Registered           Name
1    applet  system  ioswdsysmon         Off   Fri Jul 23 02:27:28 2004  IOSWD_Sample1
 sub1 cpu_util {taskname {IP Input} op ge val 1 period 10.000 }
 action 11 syslog msg IOSWD_Sample1 Policy Triggered

2    applet  system  ioswdsysmon         Off   Fri Jul 23 02:23:52 2004  IOSWD_Sample2
 sub1 mem_used {taskname {Net Input} op gt val 100 is_percent FALSE}
 action 11 syslog msg IOSWD_Sample2 Policy Triggered

3    applet  system  ioswdsysmon         Off   Fri Jul 23 03:07:38 2004  IOSWD_Sample3
 sub1 mem_used {taskname {IP RIB Update} op gt val 50 is_percent TRUE period 60.000 }
 action 1 syslog msg "IOSWD_Sample3 Policy Triggered"
```

Additional References

The following sections provide references related to Embedded Event Manager 2.0.

Related Documents

Related Topic	Document Title
CNS event agent	*CNS Event Agent* feature document, Release 12.2(2)T
CNS Configuration Engine	*Cisco CNS Configuration Engine Administrator Guide, 1.3*
Command references for Cisco IOS Release 12.2	Cisco IOS Command References, Release 12.2

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
CISCO-EMBEDDED-EVENT-MGR-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new and modified commands.

action cns-event

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

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

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

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
msg	Specifies the message to be sent to the CNS Event Bus. • *msg-text*—Character text, an environment variable, or a combination of the two. If the string contains embedded blanks, enclose it in double quotation marks.

Defaults

No messages are sent to the CNS Event Bus.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Examples

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

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

Related Commands

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

action counter

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

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

no action *label* **counter name** *counter-name*

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
name	Specifies the name of the counter to be set or modified. • *counter-name*—Name of the counter to be set or modified. The counter name is referenced in a registered counter type policy.
value	Specifies the value to be used to set or modify the counter. • *counter-value*—Number in the range from –2147483648 to 2147483647 inclusive.
op	Indicates the operator to be used with the *counter-value* to set or modify the specified counter. One of the following operators must be specified: • **dec**—If the **dec** operator is specified, the counter is decreased in value by the amount specified in the *counter-value* argument. • **inc**—If the **inc** operator is specified, the counter is increased in value by the amount specified in the *counter-value* argument. • **nop**—If the **nop** operator is specified, the counter value is read from the environment variable $_counter_value_remain. • **set**—If the **set** operator is specified, the counter is set to the value specified in the *counter-value* argument.

Defaults

No counter values are modified.

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

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

Examples

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

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

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

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

Related Commands

Command	Description
event counter	Specifies the event criteria for an EEM applet that is run on the basis of a named counter crossing a threshold.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

action force-switchover

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

action *label* **force-switchover**

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

Syntax Description

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

Defaults

A switch to a secondary processor is not made.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

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

Examples

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

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 2.0 force-switchover
```

Related Commands

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

action publish-event

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

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

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

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
sub-system	Specifies an identifier for the subsystem named in the *sub-system-id* argument that will publish the application event. • *sub-system-id*—Identifier of the subsystem. Number in the range from 1 to 4294967295. If the event is to be published by an EEM policy, the *sub-system-id* reserved for a customer policy is 798.
type	Specifies the value of an event type within the specified event. • *event-type*—Event type value. Number in the range from 1 to 4294967295.
arg1	Specifies the argument data to be passed to the application-specific event when the event is published. • *argument-data*—Character text, an environment variable, or a combination of the two.
arg2 **arg3** **arg4**	(Optional) Specifies the argument data to be passed to the application-specific event when the event is published. • *argument-data*—Character text, an environment variable, or a combination of the two.

Defaults

No application-specific events are published.

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Examples

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

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

Related Commands

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

action reload

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

action *label* **reload**

no action *label* **reload**

Syntax Description

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

Defaults

No reload of the Cisco IOS software is performed.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

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

Examples

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

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 3.0 reload
```

Related Commands

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

action snmp-trap

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

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

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

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
intdata1	(Optional) Specifies an integer to be sent in the SNMP trap message to the SNMP agent. • *integer*—Integer.
intdata2	(Optional) Specifies a second integer to be sent in the SNMP trap message to the SNMP agent. • *integer*—Integer.
strdata	(Optional) Specifies a string to be sent in the SNMP trap message to the SNMP agent. • *string*—A series of characters up to 256 characters in length. If the string contains embedded blanks, enclose it in double quotation marks.

Defaults

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

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

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

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

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

Examples

The following example shows an EEM applet called IPSLAping1 being registered to run when there is an exact match on the value of a specified SNMP object ID that represents a successful IP SLA ICMP echo operation (this is equivalent to a **ping** command). Four actions are triggered when the echo operation fails, and event monitoring is disabled until after the second failure. A message 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
Router(config-applet)# action 1.3 counter name _IPSLA1F value 1 op inc
```

Related Commands

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

action syslog

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

action *label* **syslog** [**priority** *priority-level*] **msg** *msg-text*

no action *label* **syslog** [**priority** *priority-level*] **msg** *msg-text*

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
priority	(Optional) Specifies the priority level of the syslog messages. If this keyword is selected, the *priority-level* argument must be defined. If this keyword is not selected, all syslog messages are set at the informational priority level. • *priority-level*—Number or name of the desired priority level at which syslog messages are set. Priority levels are as follows (enter the number or the keyword): – {**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. This is the default. – {**7** | **debugging**}—Debugging messages.
msg	Specifies the message to be logged. • *msg-text*—Character text, an environment variable, or a combination of the two. If the string contains embedded blanks, enclose it in double quotation marks. **Note** Messages written to syslog from an EEM applet are not screened for EEM syslog events, which may lead to recursive EEM syslog events. Messages sent from an EEM applet include the applet name for identification.

Defaults

No messages are written to syslog.

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Examples

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

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 4.0 syslog msg "Memory exhausted; current available memory
is $_snmp_oid_val bytes"
```

Related Commands

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

debug event manager

To turn on the debugging output of Embedded Event Manager (EEM) processes, use the **debug event manager** command in privileged EXEC mode. To turn off debugging output, use the **no** form of this command or the **undebug** command.

debug event manager {**action cns** | **all** | **api calls** | **api errors** | **detector application** | **detector counter** | **detector interface** | **detector ioswdsysmon** | **detector memory** | **detector rf** | **detector snmp** | **detector syslog** | **detector timer** | **policydir** | **server events** | **server scheduling** | **tcl cli_library** | **tcl commands** | **tcl smtp_library**}

no debug event manager {**action cns** | **all** | **api calls** | **api errors** | **detector application** | **detector counter** | **detector interface** | **detector ioswdsysmon** | **detector memory** | **detector rf** | **detector snmp** | **detector syslog** | **detector timer** | **policydir** | **server events** | **server scheduling** | **tcl cli_library** | **tcl commands** | **tcl smtp_library**}

Syntax Description

action cns	Displays debugging messages about CNS event messages.
all	Displays all debugging messages.
api calls	Displays debugging messages about application programming interface (API) calls.
api errors	Displays debugging messages about API errors.
detector application	Displays debugging messages about the application-specific event detector.
detector counter	Displays debugging messages about the counter event detector.
detector interface	Displays debugging messages about the interface counter event detector.
detector ioswdsysmon	Displays debugging messages about the watchdog event detector.
detector memory	Displays debugging messages about the memory thresholding event detector.
detector rf	Displays debugging messages about the redundancy event detector.
detector snmp	Displays debugging messages about the Simple Network Management Protocol (SNMP) event detector.
detector syslog	Displays debugging messages about the syslog event detector.
detector timer	Displays debugging messages about the timer event detector.
policydir	Displays debugging messages about the EEM policy director.
server events	Displays debugging messages about the EEM server events.
server scheduling	Displays all debugging messages about the EEM server scheduling events.
tcl cli_library	Displays all debugging messages about the Tool Command Language (Tcl) command-line interface (CLI) library.
tcl commands	Displays all debugging messages about the Tcl commands.
tcl smtp_library	Displays all debugging messages about the Tcl Simple Mail Transfer Protocol (SMTP) library.

Command Modes

Privileged EXEC

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.

Usage Guidelines

Use the **debug event manager** command to troubleshoot EEM command operations. Use any debugging command with caution as the volume of generated output can slow or stop the router operations. We recommend that this command be used only under the supervision of a Cisco engineer.

Examples

The following example turns on debugging messages about EEM server events and then configures an applet to write a message—Test message—to syslog. The debug output that follows displays the various EEM operations that occur as the applet is processed.

```
Router# debug event manager server events

Debug Embedded Event Manager server events debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# event manager applet timer-test
Router(config-applet)# event timer countdown time 20
Router(config-applet)# action label1 syslog msg "Test message"
Router(config-applet)# end
Router#

03:46:55: fh_server: fh_io_msg: received msg 6 from client jobid 11
03:46:55: fh_server: fh_io_msg: handling event register with esid = 23
03:46:55: fh_msg_send_to_fd: receive a reply msg, minor: 5
03:46:55: fh_server: fh_io_msg: received msg 26 from client jobid 11
03:46:55: fh_msg_send_to_fd: receive a reply msg, minor: 5
03:46:55: %SYS-5-CONFIG_I: Configured from console by console
03:47:15: fd_pulse_hndlr: received a pulse from /dev/fm/fd_timer
03:47:15: fh_msg_send_to_fd: receive a reply msg, minor: 5
03:47:15: fd_pulse_hndlr: received FH_MSG_EVENT_PUBLISH
03:47:15: fh_schedule_callback: fh_schedule_callback: cc=632C0B68 prev_epc=0; epc=63A41670
03:47:15: fh_io_msg: received FH_MSG_API_INIT; jobid=13, processid=82, client=3, job
name=EEM Callback Thread
03:47:15: fh_server: fh_io_msg: received msg 10 from client jobid 13
03:47:15: %HA_EM-6-LOG: timer-test: Test message
03:47:15: fh_server: fh_io_msg: received msg 62 from client jobid 13
03:47:15: fh_schedule_callback: fh_schedule_callback: cc=632C0B68 prev_epc=63A41670; epc=0
03:47:15: fh_server: fh_io_msg: received msg 1 from client jobid 13
03:47:15: fh_io_msg: received FH_MSG_API_CLOSE client=3
```

event application

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

event application sub-system *sub-system-id* **type** *event-type*

no event application sub-system *sub-system-id* **type** *event-type*

Syntax Description

sub-system	Specifies an identifier for the subsystem named in the *sub-system-id* argument that will publish the application event. • *sub-system-id*—Identifier of the subsystem. Number in the range from 1 to 4294967295. If the event is to be published by an EEM policy, the *sub-system-id* reserved for a policy is 798.
type	Specifies the value of an event type within the specified event. • *event-type*—Event type value. Number in the range from 1 to 4294967295.

Defaults

No EEM event criteria are specified.

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

Examples

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

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

Related Commands

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

event counter

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

event counter name *counter-name* **entry-op** *operator* **entry-val** *entry-value* [**exit-op** *operator*] [**exit-val** *exit-value*]

no event counter name *counter-name* **entry-op** *operator* **entry-val** *entry-value* [**exit-op** *operator*] [**exit-val** *exit-value*]

Syntax Description

name	Specifies that the counter named in the *counter-name* argument will be monitored. • *counter-name*—Name of the counter.
entry-op	Compares the contents of the current counter value with the entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met. 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 counter are compared to decide if a counter event should be raised. • *entry-value*—Entry counter value. Number in the range from –2147483648 to 2147483647, inclusive.
exit-op	(Optional) Compares the contents of the current counter with the exit value using the specified operator. If there is a match, an event is triggered and event monitoring is reenabled. 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.
exit-val	(Optional) Specifies the value with which the contents of the current counter are compared to decide whether the exit criteria are met. • *exit-value*—Exit counter value. Number in the range from –2147483648 to 2147483647, inclusive.

Defaults

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

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

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

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

Examples

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

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

Related Commands

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

event interface

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

event interface name *interface-type interface-number* **parameter** *counter-name* **entry-op** *operator* **entry-val** *entry-value* **entry-val-is-increment** {**true** | **false**} [**exit-comb** {**or** | **and**}] [**exit-op** *operator* **exit-val** *exit-value*] [**exit-val-is-increment** {**true** | **false**}] [**exit-time** *exit-time-value*] **poll-interval** *poll-int-value*

no event interface name *interface-type interface-number* **parameter** *counter-name* **entry-op** *operator* **entry-val** *entry-value* **entry-val-is-increment** {**true** | **false**} [**exit-comb** {**or** | **and**}] [**exit-op** *operator* **exit-val** *exit-value*] [**exit-val-is-increment** {**true** | **false**}] [**exit-time** *exit-time-value*] **poll-interval** *poll-int-value*

Syntax Description

name	Specifies the type and number of the interface to monitor. • *interface-type interface-number*—Interface type and number. For example, FastEthernet 0/1.
parameter	Specifies the name of the counter to monitor. Supported values for the *counter-name* argument are one of the following: • **input_errors**—Includes runts, giants, no buffer, cyclic redundancy checksum (CRC), frame, overrun, and ignored counts. Other input-related errors can also cause the input errors count to be increased. Some datagrams may have more than one error. • **input_errors_crc**—Number of packets with a CRC generated by the originating LAN station or remote device that does not match the checksum calculated from the data received. • **input_errors_frame**—Number of packets received incorrectly that have a CRC error and a noninteger number of octets. • **input_errors_overrun**—Number of times the receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data. • **input_packets_dropped**—Number of packets dropped because of a full input queue. • **interface_resets**—Number of times an interface has been completely reset. • **output_buffer_failures**—Number of failed buffers and number of buffers swapped out. • **output_buffer_swappedout**—Number of packets swapped to DRAM. • **output_errors**—Sum of all errors that prevented the final transmission of datagrams out of the interface being examined. Note that this may not balance with the sum of the output errors because some datagrams may have more than one error and other datagrams may have errors that do not fall into any of the specifically tabulated categories. • **output_errors_underrun**—Number of times that the transmitter has been running faster than the router can handle. • **output_packets_dropped**—Number of packets dropped because of a full output queue. • **receive_broadcasts**—Number of broadcast or multicast packets received by the interface. • **receive_giants**—Number of packets that are discarded because they exceed the maximum packet size of the medium. • **receive_rate_bps**—Interface receive rate, in bytes per second. • **receive_rate_pps**—Interface receive rate, in packets per second. • **receive_runts**—Number of packets that are discarded because they are smaller than the minimum packet size of the medium. • **receive_throttle**—Number of times the receiver on the port was disabled, possibly because of buffer or processor overload.

	• **reliability**—Reliability of the interface, as a fraction of 255 (255 out of 255 is 100 percent reliability), calculated as an exponential average over 5 minutes. • **rxload**—Receive rate of the interface, as a fraction of 255 (255 out of 255 is 100 percent). • **transmit_rate_bps**—Interface transmit rate, in bytes per second. • **transmit_rate_pps**—Interface transmit rate, in packets per second. • **txload**—Transmit rate of the interface, as a fraction of 255 (255 out of 255 is 100 percent).
entry-op	Compares the current interface counter value with the entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met. 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 current interface counter value is compared to decide if the interface event should be raised. • *entry-value*—Entry value. Number in the range from –2147483648 to 2147483647, inclusive.
entry-val-is-increment	Indicates whether the *entry-value* is an absolute or an increment value. • If the **true** keyword is specified, the *entry-value* is an increment and the interface event is raised whenever the increment value occurs. • If the **false** keyword is specified, the *entry-value* is an absolute value and the interface event is raised whenever the absolute value occurs. This is the default.
exit-comb	(Optional) Indicates the combination of exit conditions that must be met before event monitoring is reenabled. • If the **or** operator is specified, an exit comparison operator and an exit object ID value, or an exit time value must exist. • If the **and** operator is specified, 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 interface counter value with the exit value using the specified operator. If there is a match, an event is triggered and event monitoring is reenabled. 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.
exit-val	(Optional) Specifies the value with which the contents of the current interface counter value are compared to decide whether the exit criteria are met. If an exit value is specified, you must configure an exit operator. • *exit-value*—Exit value. Number in the range from –2147483648 to 2147483647, inclusive.
exit-val-is-increment	(Optional) Indicates whether the *exit-value* is an absolute or an increment value. • If the **true** keyword is specified, the *exit-value* is an increment and the event monitoring is reenabled whenever the increment value occurs. • If the **false** keyword is specified, the *exit-value* is an absolute value and the event monitoring is reenabled whenever the absolute value occurs. This is the default.
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*—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.
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. If using milliseconds, specify the milliseconds in the format 1.mmm. The minimum polling interval is 1 second.

Defaults

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

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

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.

Examples

The following example shows how a policy named EventInterface is triggered every time the receive_throttle counter for the FastEthernet0/0 interface in incremented by 5. The polling interval to check the counter is specified to run once every 10 seconds.

```
Router(config)# event manager applet EventInterface
Router(config-applet)# event interface name FastEthernet0/0 parameter receive_throttle
entry-op ge entry-val 5 entry-val-is-increment true poll-interval 10
Router(config-applet)# action 1.0 syslog msg "Applet EventInterface"
```

Related Commands

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

event ioswdsysmon

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

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

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

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

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

mem-proc taskname *process-name* **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 two forms of syntax shown above under the Subevent Syntax heading.
timewin	(Optional) Specifies the time window within which all of the subevents must occur in order for an event to be generated. • *timewin-value*—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. • If the **and** operator is specified, both the results of subevent 1 and subevent 2 must cross the specified thresholds. • If the **or** operator is specified, the results of either subevent 1 or subevent 2 must cross the specified thresholds. • If the **andnot** operator is specified, only the results from subevent 1 must cross the specified threshold.
sub2	(Optional) Specifies the second subevent. • *subevent2*—Second subevent. Use one of the two forms of syntax shown above under the Subevent Syntax heading.
cpu-proc	Specifies the use of a sample collection of CPU statistics.
mem-proc	Specifies the use of a sample collection of memory statistics.
taskname	Specifies a process name. • *process-name*—Name of the Cisco IOS process to be monitored. If the process name contains embedded blanks, enclose it in double quotation marks.

op	Compares the collected CPU or memory usage sample with the value specified in the *value* argument. If there is a match, the subevent is triggered. The *operator* argument takes one of the following values: • **gt**—Greater than. • **ge**—Greater than or equal to. • **eq**—Equal to. • **ne**—Not equal to. • **lt**—Less than. • **le**—Less than or equal to.
val	Specifies the value with which the collected CPU or memory usage sample is compared to decide if the subevent should be raised. • *value*—Number in the range from 1 to 4294967295.
period	(Optional) Specifies the elapsed time period for the collection samples to be averaged. • *period-value*—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. If the time period is 0, the most recent sample is used.
is-percent	(Optional) Indicates whether the *value* argument is a percentage. • If the **true** keyword is specified, the *value* argument is a percentage. • If the **false** keyword is specified, the *value* argument is not a percentage.

Defaults

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

Command Modes

Applet configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

Examples

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

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

Related Commands

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

event manager applet

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

event manager applet *applet-name*

no event manager applet *applet-name*

Syntax Description

applet-name	Name of the applet file.

Defaults

No EEM applets are registered.

Command Modes

Global configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Usage Guidelines

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

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

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

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

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

Examples

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

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

Related Commands

Command	Description
show event manager policy registered	Displays registered Embedded Event Manager policies.

event manager environment

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

event manager environment *variable-name string*

no event manager environment *variable-name*

Syntax Description

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

Defaults

No EEM environment variables are set.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

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

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

Examples

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

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

Related Commands

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

event manager history size

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

event manager history size {**events** | **traps**} [*size*]

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

Syntax Description

events	Changes the size of the EEM event history table.
traps	Changes the size of the EEM Simple Network Management Protocol (SNMP) trap history table.
size	(Optional) Number of history table entries. Range is from 1 to 50. Default is 50.

Defaults

The size of the history table is 50 entries.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Examples

The following example of the **event manager history size** command changes the size of the SNMP trap history table to 30 entries:

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

Related Commands

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

event manager policy

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

event manager policy *policy-filename* [**type system**] [**trap**]

no event manager policy *policy-filename*

Syntax Description

policy-filename	Name of the policy file.
type	(Optional) Specifies the type of EEM policy to be registered.
system	(Optional) Registers a Cisco-defined system policy.
trap	(Optional) Generates a Simple Network Management Protocol (SNMP) trap when the policy is triggered.

Defaults

No EEM policies are registered.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

If you enter the **event manager policy** command without specifying the optional **type** keyword, the EEM tries to locate the specified policy file in the system policy directory. If the EEM finds the file in the system policy directory, it registers the policy as a system policy. If the EEM does not find the specified policy file in the system policy directory, the policy is not registered.

Examples

The following example of the **event manager policy** command registers a system-defined policy named tm_countdown_ios.tcl located in the system policy directory:

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

Related Commands

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

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.

Defaults

Policy scheduling is active.

Command Modes

Global configuration

Command History

Release	Modification
12.2(25)S	This command was introduced.

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.

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 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 snmp oid *oid-value* **get-type** {**exact** | **next**} **entry-op** *operator* **entry-val** *entry-value* [**exit-comb** {**or** | **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-time** *exit-time-value*] **poll-interval** *poll-int-value*

no event snmp oid *oid-value* **get-type** {**exact** | **next**} **entry-op** *operator* **entry-val** *entry-value* [**exit-comb** {**or** | **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-time** *exit-time-value*] **poll-interval** *poll-int-value*

Syntax Description

oid	Specifies the SNMP object identifier (object ID) values 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. 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. 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.
exit-comb	(Optional) Indicates the combination of exit conditions that must be met before event monitoring is reenabled. • If the **or** operator is specified, an exit comparison operator and an exit object ID value, or an exit time value must exist. • If the **and** operator is specified, 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. 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.
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*—Exit object ID value of the data element.
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*—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.
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. If using milliseconds, specify the milliseconds in the format 1.mmm. The minimum polling interval is 1 second.

Defaults

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

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

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.

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
Router(config-applet)# action 1.3 counter name _IPSLA1F value 1 op inc
```

Related Commands

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

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 syslog [**occurs** *num-occurrences*] [**period** *period-value*] [**priority** *priority-level*] **pattern** *regular-expression*

no event syslog [**occurs** *num-occurrences*] [**period** *period-value*] [**priority** *priority-level*] **pattern** *regular-expression*

Syntax Description

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*—The number of occurrences. The value must be greater than 0.
period	(Optional) Specifies the time interval during which the one or more occurrences must take place. If the keyword is not specified, no time period check is applied. • *period-value*—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.
priority	(Optional) Specifies the priority level of the syslog messages to be screened. If the keyword is selected, the *priority-level* argument must be defined. If the keyword is not specified, the software will use the default of **priority all**, and all priorities will be considered when log messages are scanned. • *priority-level*—The number or name of the desired priority level at which syslog messages are matched. Messages at or numerically lower than the specified level are matched. Priority levels are as follows (enter the keyword or number, if available): – **all**—All priorities are considered when log messages are scanned. – {**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.
pattern	Specifies the regular expression used to perform the syslog message pattern match. • *regular-expression*—Regular expression.

Defaults

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

Command Modes

Applet configuration

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

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.

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 occurs 4 pattern {.*UPDOWN.*Ethernet1/0.*}
Router(config-applet)# action 1.0 syslog msg "Ethernet interface 1/0 is down"
```

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

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

Syntax Description

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*—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.
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*—A text string that consists of five fields. The fields represent the time and date when CRON timer events will be triggered; the fields are separated by spaces. Fields and corresponding values are as follows: – *minute*—Minute when a CRON timer event is triggered. Valid entries are numbers in the range from 0 to 59. – *hour*—Hour when a CRON timer event is triggered. Valid entries are numbers in the range from 0 to 23. – *day of month*—Day of the month when a CRON timer event is triggered. Valid entries are numbers in the range from 1 to 31. – *month*—Month when a CRON timer event is triggered. Valid entries are numbers in the range from 1 to 12 or the first three letters (not case-sensitive) of the name of the month. – *day of week*—Day of the week when a CRON timer event is triggered. Valid entries are numbers in the range from 0 to 6 (Sunday is 0) or the first three letters (not case-sensitive) of the name of the day. **Note** Ranges of numbers are allowed. The specified range is inclusive, and the two numbers are separated by a hyphen. For example, 8-11 after the hour field specifies execution of a CRON timer event at hours 8, 9, 10, and 11. **Note** A field may contain an asterisk, *, which means that a field is not specified and can be any value. **Note** Lists are permitted. 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. **Note** Step values are permitted in conjunction with ranges. 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 first five fields, some special strings can be entered. See the "Usage Guidelines" section for more 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 the name of the timer. • *timer-name*—Name of the timer.

Defaults

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.

Usage Guidelines

Instead of the five fields of a UNIX crontab entry as described in the syntax description table 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 Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3.

Examples

The following example shows how to specify that an event is triggered one time after five 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 six minutes and six 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 five 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.

show event manager environment

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

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

Syntax Description

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

Defaults

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

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.

Examples

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

```
Router# show event manager environment

No.  Name                          Value
1    _cron_entry                   0-59/1 0-23/1 * * 0-7
2    _show_cmd                     show version
3    _syslog_pattern               .*UPDOWN.*Ethernet1/0.*
4    _config_cmd1                  interface Ethernet1/0
5    _config_cmd2                  no shutdown
```

Table 2 describes the significant fields shown in the display.

Table 2 show event manager environment Field Descriptions

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

Related Commands

Command	Description
event manager environment	Sets an EEM environment variable.

show event manager history events

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

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

Syntax Description

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

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

Examples

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

```
Router# show event manager history events

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

Table 3 describes the significant fields shown in the display.

Table 3 show event manager history events Field Descriptions

Field	Description
No.	Event number.
Time of Event	Date and time when the event was triggered.
Event Type	Type of event.
Name	Name of the policy that was triggered.

Related Commands

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

show event manager history traps

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

show event manager history traps {**server** | **policy**}

Syntax Description

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

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

Examples

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

```
Router# show event manager history traps policy

No.  Time                      Trap Type           Name
1    Wed Aug18  22:30:58 2004  policy              EEM Policy Director
2    Wed Aug18  22:34:58 2004  policy              EEM Policy Director
3    Wed Aug18  22:51:18 2004  policy              EEM Policy Director
```

Table 4 describes the significant fields shown in the display.

Table 4 show event manager history traps Field Descriptions

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

Related Commands

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

show event manager policy available

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

show event manager policy available [**system**]

Syntax Description

system	(Optional) Displays all available system policies.

Defaults

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

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

Examples

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

```
Router# show event manager policy available

No.  Type    Time Created                 Name
1    system  Tue Sep 12 09:41:32 2002     sl_pattern_ios.tcl
2    system  Tue Sep 12 09:41:32 2002     tm_countdown_ios.tcl
```

Table 5 describes the significant fields shown in the display.

Table 5 show event manager policy available Field Descriptions

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

Related Commands

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

show event manager policy pending

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

show event manager policy pending

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

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

Examples

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

```
Router# show event manager policy pending

No.  Time of Event             Event Type          Name
1    Sat Oct11  05:02:41 2003  timer watchdog      script:fd_timer_watchdog.tcl
2    Sat Oct11  05:02:41 2003  timer watchdog      script:fd_timer_watchdog2.tcl
```

Table 6 describes the significant fields shown in the display.

Table 6 show event manager policy pending Field Descriptions

Field	Description
No.	Index number automatically assigned to the policy.
Time of Event	Date and time when the policy was queued for execution in the EEM server.
Event Type	Type of event.
Name	Name of the EEM policy file.

Related Commands

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

show event manager policy registered

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

show event manager policy registered [**event-type** *event-name*] [**system**] [**time-ordered** | **name-ordered**]

Syntax Description

event-type	(Optional) Displays the registered policies for the event type specified in the *event-name* argument. If the event type is not specified, all registered policies are displayed. The *event-name* argument can be one of the following options: • **application**—Application event type. • **counter**—Counter event type. • **interface**—Interface event type. • **ioswdsysmon**—watchdog system monitor event type. • **snmp**—Simple Network Management Protocol (SNMP) event type. • **syslog**—Syslog event type. • **timer-absolute**—Absolute timer event type. • **timer-countdown**—Countdown timer event type. • **timer-cron**—Clock daemon (CRON) timer event type. • **timer-watchdog**—Watchdog timer event type.
system	(Optional) Displays the registered system policies.
time-ordered	(Optional) Displays the policies by the time at which they were registered. This is the default.
name-ordered	(Optional) Displays the policies in alphabetical order by policy name.

Defaults

If this command is invoked with no optional keywords, it displays the registered EEM policies for all event types. The policies are displayed according to the time at which they were registered.

Command Modes

Privileged EXEC

CommandHistory

Release	Modification
12.2(25)S	This command was introduced.

Usage Guidelines

The output of this command is most helpful to the person who has the task of writing and monitoring EEM policies. The output shows registered policy information in two parts. The first line in each policy description lists the index number assigned to the policy, the policy type (system), the type of event registered, the time when the policy was registered, and the name of the policy file. The remaining lines

of each policy description display information about the registered event and how the event is to be handled; the information comes directly from the Tool Command Language (Tcl) command arguments that make up the policy file.

Examples

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

```
Router# show event manager policy registered

No.  Class   Type    Event Type          Trap  Time Registered           Name
1    applet  system  snmp                Off   Fri Aug 13 17:42:52 2004  IPSLAping1
 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 5.000
 action 1.0 syslog priority critical msg Server IPecho Failed: OID=$_snmp_oid_val
 action 1.1 snmp-trap strdata EEM detected server reachability failure to 10.1.88.9
 action 1.2 publish-event sub-system 88000101 type 1 arg1 10.1.88.9 arg2 IPSLAEcho arg3
fail
 action 1.3 counter name _IPSLA1F value 1 op inc
```

Table 7 describes the significant fields shown in the display.

Table 7 show event manager policy registered Field Descriptions

Field	Description
No.	Index number automatically assigned to the policy.
Class	Indicates the class of policy, either applet or script.
Type	Indicates whether the policy is a system policy.
Event Type	Indicates the type of event.
Trap	Indicates whether an SNMP trap is enabled.
Time Registered	Time stamp indicating the date and time when the policy file was registered.
Name	Name of the EEM policy file.

Related Commands

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

Glossary

EEM—Embedded Event Manager. EEM is a distributed, scalable, and customized approach to event detection and recovery offered directly in a Cisco IOS device.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol such as SNMP.

SNMP—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices and to manage configurations, statistics collection, performance, and security.

Tcl—Tool Command Language. Tcl is a general-purpose programming language originally intended to be embedded in other applications as a configuration and extension language.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.