CHAPTER 35

Configuring Embedded Event Manager

Embedded Event Manager (EEM) is a distributed and customized approach to event detection and recovery within a Cisco IOS device. EEM offers the ability to monitor events and take informational, corrective, or any other EEM action when the monitored events occur or when a threshold is reached. An EEM policy defines an event and the actions to be taken when that event occurs.

This chapter tells how to use EEM and how to configure it on a Catalyst 3750-X or 3560-X switch. Unless otherwise noted, the term *switch* refers to a standalone switch or a Catalyst 3750-X switch stack.

Note Beginning with Cisco IOS Release 12.2(55)SE, the EEM feature is supported on the IP base and IP services feature set. It is not supported on the LAN base feature set.

For complete syntax and usage information for the commands used in this chapter, see the switch command reference for this release and the *Cisco IOS Network Management Command Reference.* For the complete EEM document set, see these documents in the *Cisco IOS Network Management Configuration Guide*:

- *Embedded Event Manager Overview*

  http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_eem_overview.html
- *Writing Embedded Event Manager Policies Using the Cisco IOS CLI*

  http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_eem_policy_cli.html
- *Writing Embedded Event Manager Policies Using Tcl*

  http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_eem_policy_tcl.html

This chapter consists of these sections:

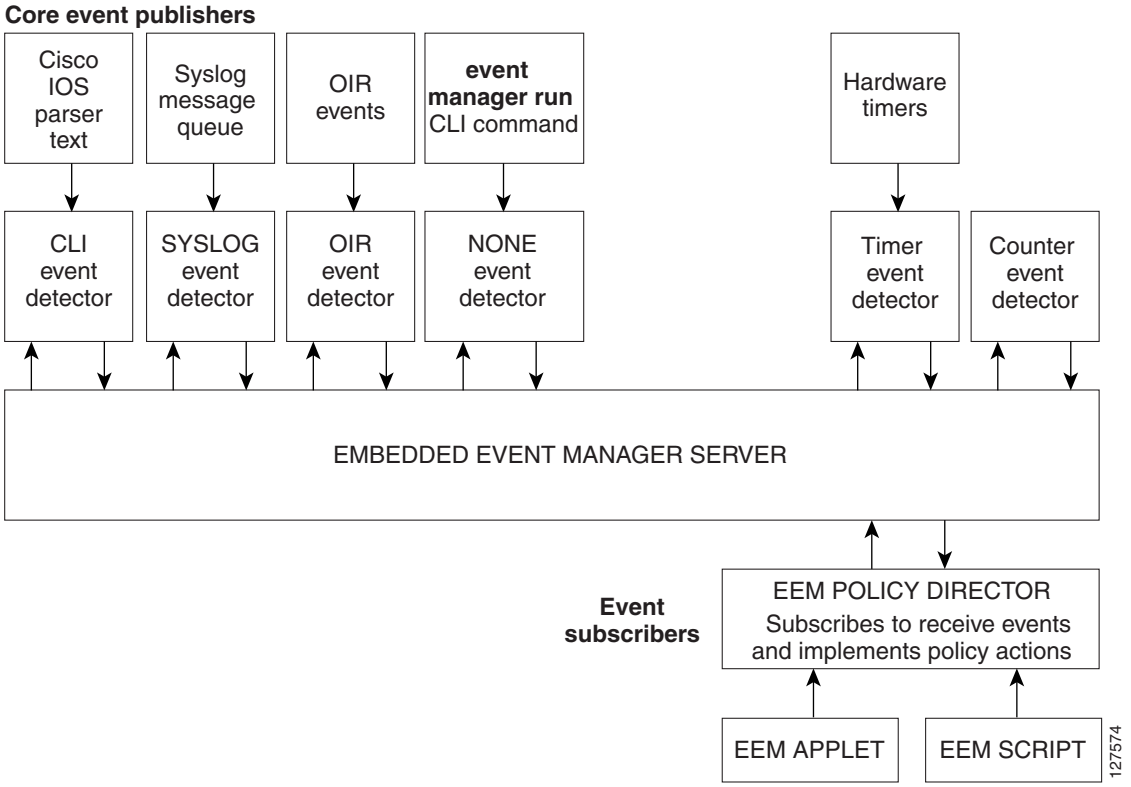
Understanding Embedded Event Manager

EEM monitors key system events and then acts on them through a set policy. This policy is a programmed script that you can use to customize a script to invoke an action based on a given set of events occurring. The script generates actions such as generating custom syslog or Simple Network Management Protocol (SNMP) traps, invoking CLI commands, forcing a failover, and so forth. The event management capabilities of EEM are useful because not all event management can be managed from the switch and

because some problems compromise communication between the switch and the external network management device. Network availability is improved if automatic recovery actions are performed without rebooting the switch.

Figure 35-1 shows the relationship between the EEM server, the core event publishers (event detectors), and the event subscribers (policies). The event publishers screen events and when there is a match on an event specification that is provided by the event subscriber. Event detectors notify the EEM server when an event occurs. The EEM policies then implement recovery based on the current state of the system and the actions specified in the policy for the given event.

Figure 35-1 Embedded Event Manager Core Event Detectors



See the *EEM Configuration for Cisco Integrated Services Router Platforms Guide* for examples of EEM deployment.

- Event Detectors, page 35-3
- Embedded Event Manager Actions, page 35-4
- Embedded Event Manager Policies, page 35-4
- Embedded Event Manager Environment Variables, page 35-5
- EEM 3.2, page 35-5

Event Detectors

EEM software programs known as event detectors determine when an EEM event occurs. Event detectors are separate systems that provide an interface between the agent being monitored, for example SNMP, and the EEM polices where an action can be implemented. Event detectors are generated only by the master switch. CLI and routing processes also run only from the master switch.

Note The stack member switch does not generate events and does not support memory threshold notifications or IOSWdSysmon event detectors.

EEM allows these event detectors:

- Application-specific event detector—Allows any EEM policy to publish an event.
- IOS CLI event detector—Generates policies based on the commands entered through the CLI.
- Generic Online Diagnostics (GOLD) event detector—Publishes an event when a GOLD failure event is detected on a specified card and subcard.
- Counter event detector—Publishes an event when a named counter crosses a specified threshold.
- 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.For example, if the incremental value is set to 50 an event would be published when the interface counter increases by 50.

  This detector also publishes an event about an interface based on the rate of change for the entry and exit values.
- None event detector—Publishes an event when the **event manager run** CLI command executes an EEM policy. EEM schedules and runs policies on the basis on an event specification within the policy itself. An EEM policy must be manually identified and registered before the **event manager run** command executes.
- Online insertion and removal event detector—Publishes an event when a hardware insertion or removal (OIR) event occurs.
- Resource threshold event detector—Generates policies based on global platform values and thresholds. Includes resources such as CPU utilization and remaining buffer capacity. Applies only to the master switch.
- Remote procedure call (RPC) event detector—Invokes EEM policies from outside the switch over an encrypted connecting using Secure Shell (SSH) and uses Simple Object Access Protocol (SOAP) data encoding for exchanging XML-based messages. It also runs EEM policies and then gets the output in a SOAP XML-formatted reply.
- 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.
  - The object matches specified values or crosses specified thresholds.
  - The SNMP delta value, the difference between the monitored Object Identifier (OID) value at the beginning the period and the actual OID value when the event is published, matches a specified value.
- SNMP notification event detector—Intercepts SNMP trap and inform messages received by the switch. The event is generated when an incoming message matches a specified value or crosses a defined threshold.

- 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—Publishes events for
  - 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. The timer automatically resets itself to its initial value and starts to count down again.
  - A CRON timer publishes an event by using a UNIX standard CRON specification to define when the event is to be published. A CRON timer never publishes events more than once per minute.
- Watchdog event detector (IOSWDSysMon)—Publishes an event only on the master switch when

  Publishes an event when one of these events occurs:
  - CPU utilization for a Cisco IOS process crosses a threshold.
  - Memory utilization for a Cisco IOS process crosses a threshold.

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

Embedded Event Manager Actions

These actions occur in response to an event:

- Modifying a named counter.
- Publishing an application-specific event.
- Generating an SNMP trap.
- Generating prioritized syslog messages.
- Reloading the Cisco IOS software.
- Reloading the switch stack.
- Reloading the master switch in the event of a master switchover. If this occurs, a new master switch is elected.

Embedded Event Manager Policies

EEM can monitor events and provide information, or take corrective action when the monitored events occur or 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 policy that is defined within the CLI configuration. It is a concise method for defining event screening criteria and the actions to be taken when that event occurs. Scripts are defined on the networking device by using an ASCII editor. The script, which can be a bytecode (.tbc) and text (.tcl) script, is then copied to the networking device and registered with EEM. You can also register multiple events in a .tcl file.

You use EEM to write and implement your own policies using the EEM policy tool command language (TCL) script. When you configure a TCL script on the master switch and the file is automatically sent to the member switches. The user-defined TCL scripts must be available in the member switches so that if the master switch changes, the TCL scripts policies continue to work.

Cisco enhancements to TCL in the form of keyword extensions facilitate the development of EEM policies. These keywords identify the detected event, the subsequent action, utility information, counter values, and system information.

For complete information on configuring EEM policies and scripts, see the *Cisco IOS Network Management Configuration Guide, Release 12.4T.*

Embedded Event Manager Environment Variables

EEM uses environment variables in EEM policies. These variables are defined in a EEM policy tool command language (TCL) script by running a CLI command and the **event manager environment** command.

User-defined variables

Defined by the user for a user-defined policy.

- Cisco-defined variables

  Defined by Cisco for a specific sample policy.

- Cisco built-in variables (available in EEM applets)

  Defined by Cisco and can be read-only or read-write. The read-only variables are set by the system before an applet starts to execute. The single read-write variable, _exit_status, allows you to set the exit status for policies triggered from synchronous events.

Cisco-defined environment variables and Cisco system-defined environment variables might 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 by using the **event manager environment** global configuration command. You must defined the variables in the EEM policy before you register the policy.

For information about the environmental variables that EEM supports, see the *Cisco IOS Network Management Configuration Guide, Release 12.4T.*

EEM 3.2

EEM 3.2 is supported in Cisco IOS Release 12.2(52)SE and later and introduces these event detectors:

- Neighbor Discovery—Neighbor Discovery event detector provides the ability to publish a policy to respond to automatic neighbor detection when:
  - a Cisco Discovery Protocol (CDP) cache entry is added, deleted, or updated.
  - a Link Layer Discovery Protocol (LLDP) cache entry is added, deleted or updated.
  - an interface link status changes.
  - an interface line status changes.
- Identity—Identity event detector generates an event when AAA authorization and authentication is successful, when failure occurs, or after normal user traffic on the port is allowed to flow.

- Mac-Address-Table—Mac-Address-Table event detector generates an event when a MAC address is learned in the MAC address table.

Note The Mac-Address-Table event detector is supported only on switch platforms and can be used only on Layer 2 interfaces where MAC addresses are learned. Layer 3 interfaces do not learn addresses ,and routers do not usually support the MAC address-table infrastructure needed to notify EEM of a learned MAC address.

EEM 3.2 also introduces CLI commands to support the applets to work with the new event detectors.

For further details about EEM 3.2 features, see the Embedded Event Manager 3.2 document.

http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_eem_3.2.html

Configuring Embedded Event Manager

For complete information about configuring embedded event manager, see the *Cisco IOS Network Management Configuration Guide, Release 12.4T.*

Note To configure EEM, you must have the IP services feature set installed on the switch.

Registering and Defining an Embedded Event Manager Applet

Beginning in privileged EXEC mode, perform this task to register an applet with EEM and to define the EEM applet using the **event applet** and **action applet** configuration commands.

Note Only one event applet command is allowed in an EEM applet. Multiple action applet commands are permitted. If you do not specify the **no event** and **no action** commands, the applet is removed when you exit configuration mode.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**event manager applet** *applet-name*	Register the applet with EEM and enter applet configuration mode.
Step 3	**event snmp oid** *oid-value* **get-type** {**exact** | **next**} **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} **entry-val** *entry-val* [**exit-comb** {**or** | **and**}] [**exit-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**}] [**exit-val** *exit-val*] [**exit-time** *exit-time-val*] **poll-interval** *poll-int-val*	Specify the event criteria that causes the EEM applet to run. (Optional) Exit criteria. If exit criteria are not specified, event monitoring is re-enabled immediately.

	Command	Purpose
Step 4	**action label syslog** [**priority** *priority-level*] **msg** *msg-text*	Specify the action when an EEM applet is triggered. Repeat this action to add other CLI commands to the applet. • (Optional) The priority keyword specifies the priority level of the syslog messages. If selected, you need to define the priority-level argument. • For *msg-text,* the argument can be character text, an environment variable, or a combination of the two.
Step 5	**end**	Exit applet configuration mode and return to privileged EXEC mode.

This example shows the output for EEM when one of the fields specified by an SNMP object ID crosses a defined threshold:

```
Switch(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
```

These examples show actions that are taken in response to an EEM event:

```
Switch(config-applet)# action 1.0 syslog priority critical msg "Memory exhausted; current
available memory is $_snmp_oid_val bytes"
Switch (config-applet)# action 2.0 force-switchover
```

Registering and Defining an Embedded Event Manager TCL Script

Beginning in privileged EXEC mode, perform this task to register a TCL script with EEM and to define the TCL script and policy commands.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**show event manager environment** [**all** | *variable-name*]	(Optional) The **show event manager environment** command displays the name and value of the EEM environment variables. (Optional) The **all** keyword displays the EEM environment variables. (Optional) The *variable-name* argument displays information about the specified environment variable.
Step 3	**configure terminal**	Enter global configuration mode.
Step 4	**event manager environment variable-name string**	Configure the value of the specified EEM environment variable. Repeat this step for all the required environment variables.
Step 5	**event manager policy policy-file-name** [**type system**] [**trap**]	Register the EEM policy to be run when the specified event defined within the policy occurs.
Step 6	**exit**	Exit global configuration mode and return to privileged EXEC mode.

This example shows the sample output for the show event manager environment command:

```
Switch# 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
```

This example shows a CRON timer environment variable, which is assigned by the software, to be set to every second minute, every hour of every day:

```
Switch(config)# event manager environment_cron_entry 0-59/2 0-23/1 * * 0-6
```

This example shows the sample EEM policy named *tm_cli_cmd.tcl* registered as a system policy. The system policies are part of the Cisco IOS image. User-defined TCL scripts must first be copied to flash memory.

```
Switch(config)# event manager policy tm_cli_cmd.tcl type system
```

Displaying Embedded Event Manager Information

To display information about EEM, including EEM registered policies and EEM history data, see the *Cisco IOS Network Management Command Reference.*