CHAPTER 7

Configuring the Embedded Event Manager

Embedded Event Manager monitors events that occur on your device and takes action to recover or troubleshoot these events, based on your configuration.

This chapter describes how to configure the EEM to detect and handle critical events on a device.

This chapter includes the following sections:

About EEM

This section includes the following topics:

EEM Overview

EEM consists of three major components:

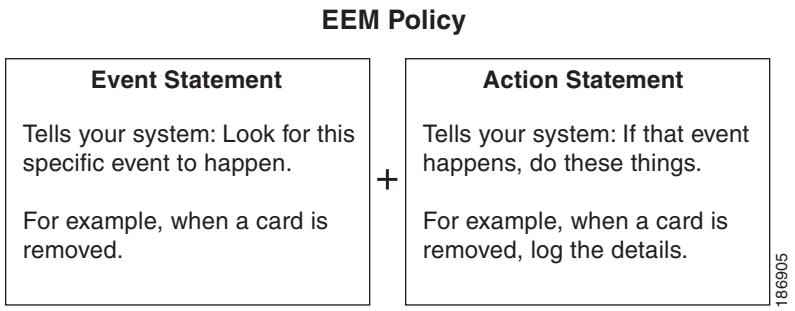
- Event statements—Events to monitor from another Cisco NX-OS component that may require some action, workaround, or notification.
- Action statements —An action that EEM can take, such as sending an e-mail, or disabling an interface, to recover from an event.
- Policies—An event paired with one or more actions to troubleshoot or recover from the event.

Policies

An EEM policy consists of an event statement and one or more action statements. The event statement defines the event to look for as well as the filtering characteristics for the event. The action statement defines the action EEM takes when the event occurs.

Figure 7-1 shows the two basic statements in an EEM policy.

Figure 7-1 EEM Policy Statements



You can configure EEM policies using the CLI or using a VSH script.

Note EEM policy matching is not supported on MDS switches.

EEM maintains event logs on the supervisor.

Cisco NX-OS has a number of preconfigured system policies. These system policies define many common events and actions for the device. System policy names begin with two underscore characters (__).

You can create user policies to suit your network. If you create a user policy, any actions in your policy occur after EEM triggers any system policy actions related to the same event as your policy. To configure a user policy, see the "Defining a User Policy Using the CLI" section on page 7-6.

You can also override some system policies. The overrides that you configure take the place of the system policy. You can override the event or the actions.

Use the **show event manager system-policy** command to view the preconfigured system policies and determine which policies that you can override.

To configure an overriding policy, see the "Overriding a Policy" section on page 7-11.

Note You should use the **show running-config eem** command to check the configuration of each policy. An override policy that consists of an event statement and no action statement triggers no action and no notification of failures.

Note Your override policy should always include an event statement. An override policy without an event statement overrides all possible events in the system policy.

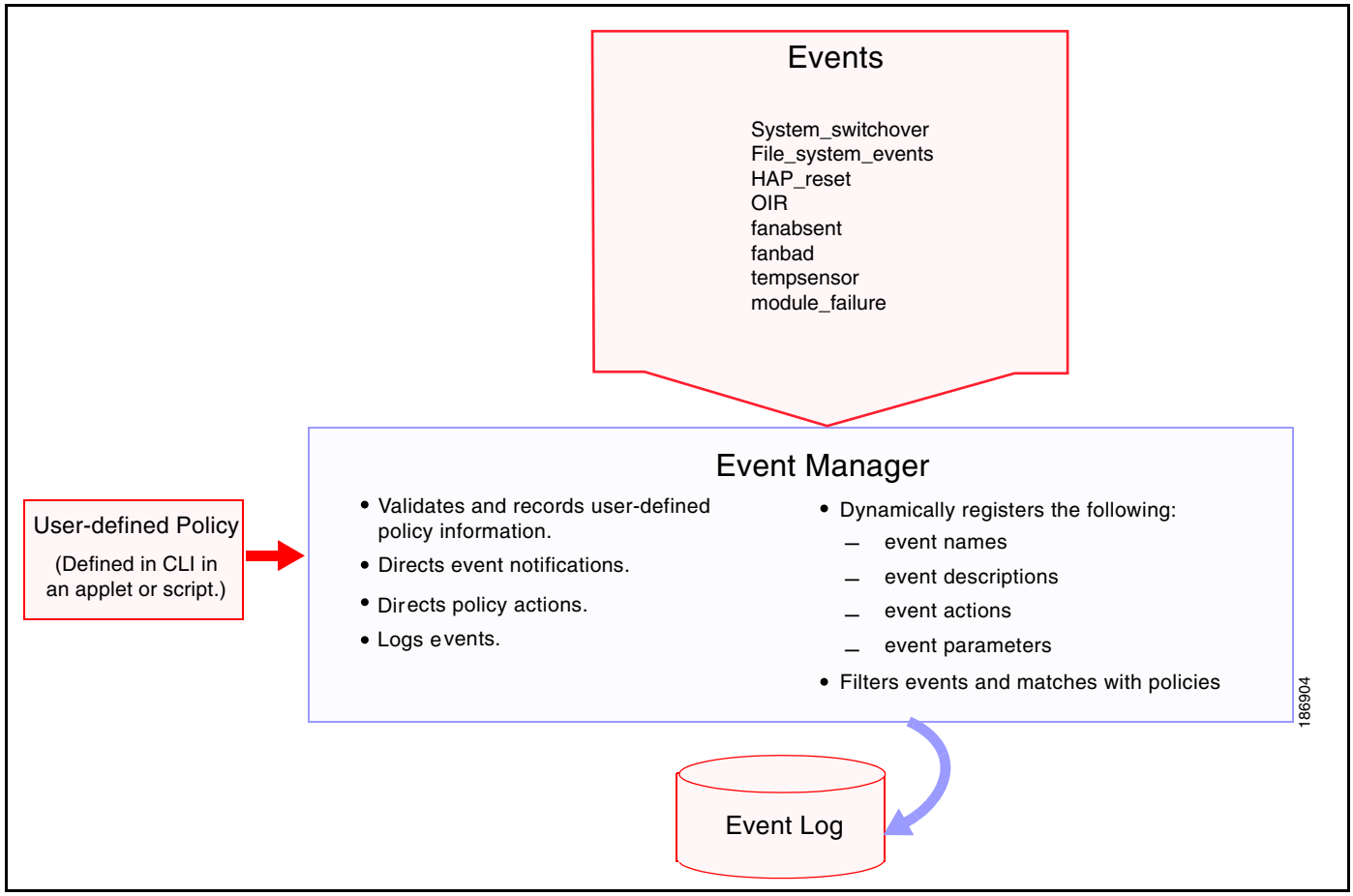
Event Statements

An event is any device activity for which some action, such as a workaround or a notification, should be taken. In many cases, these events are related to faults in the device such as when an interface or a fan malfunctions.

EEM defines event filters so only critical events or multiple occurrences of an event within a specified time period trigger an associated action.

Figure 7-2 shows events that are handled by EEM.

Figure 7-2 EEM Overview



Event statements specify the event that triggers a policy to run. You can configure only one event statement per policy.

EEM schedules and runs policies on the basis of event statements. EEM examines the event and action commands and runs them as defined.

Note If you want to allow the triggered event to process the default actions also, you must explicitly configure an EEM action with “event-default” or “policy-default”, based on the type of policy.

Action Statements

Action statements describe the action triggered by a policy. Each policy can have multiple action statements. If no action is associated with a policy, EEM still observes events but takes no actions.

EEM supports the following actions in action statements:

- Execute any CLI commands.
- Update a counter.
- Log an exception.
- Force the shut down of any module.
- Reload the device.
- Shut down specified modules because the power is over budget.
- Generate a syslog message.
- Generate a Call Home event.
- Generate an SNMP notification.
- Use the default action for the system policy.

Note If you want to allow the triggered event to process the default actions also, you must explicitly configure an EEM action with **event-default** or **policy-default**, based on the type of policy. For example, if you match a CLI command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the CLI command to execute.

Note Verify that your action statements within your user policy or overriding policy do not negate each other or adversely affect the associated system policy.

VSH Script Policies

You can also write policies in a VSH script, using a text editor. These policies have an event statement and action statement(s) just as other policies, and these policies can either augment or override system polices. After you write your script policy, copy it to the device and activate it. To configure a policy in a script, see the “Defining a Policy Using a VSH Script” section on page 7-10.

Environment Variables

You can define environment variables for EEM that are available for all policies. Environment variables are useful for configuring common values that you can use in multiple policies. For example, you can create an environment variable for the IP address of an external e-mail server.

You can use an environment variable in action statements by using the parameter substitution format.

Example 7-1 shows a sample action statement to force a module 1 shutdown, with a reset reason of "EEM action."

Example 7-1 Action Statement

```
switch (config-eem-policy)# action 1.0 forceshut module 1 reset-reason "EEM action"
```

If you define an environment variable for the shutdown reason, called default-reason, you can replace that reset reason with the environment variable, as shown in Example 7-2.

Example 7-2 Action Statement with Environment Variable

```
switch (config-eem-policy)# action 1.0 forceshut module 1 reset-reason $default-reason
```

You can reuse this environment variable in any policy. For more information on environment variables, see the "Defining an Environment Variable" section on page 7-12.

High Availability

Cisco NX-OS supports stateless restarts for EEM. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration.

Licensing Requirements for EEM

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	EEM requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you.

Prerequisites for EEM

EEM has the following prerequisites:

- You must have network-admin user privileges to configure EEM.

Configuration Guidelines and Limitations

EEM has the following configuration guidelines and limitations:

- Action statements within your user policy or overriding policy should not negate each other or adversely affect the associated system policy.
- If you want to allow the triggered event to process the default actions also, you must explicitly configure an EEM action with **event-default** or **policy-default**, based on the type of policy. For example, if you match a CLI command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the CLI command to execute.
- An override policy that consists of an event statement and no action statement triggers no action and no notification of failures.
- An override policy without an event statement overrides all possible events in the system policy.

Configuring EEM

This section includes the following topics:

Defining a User Policy Using the CLI

You can define a user policy using the CLI.

This section includes the following topics:

To define a user policy using the CLI, follow these steps:

	Command	Purpose
Step 1	**config t**	Enters configuration mode.
Step 2	**event manager applet** *applet-name*	Registers the applet with EEM and enters applet configuration mode. The *applet-name* can be any case-sensitive alphanumeric string up to 29 characters.
Step 3	**description** *policy-description*	(Optional) Configures a descriptive string for the policy. The string can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks.
Step 4	**event** *event-statement*	Configures the event statement for the policy. See the "Configuring Event Statements" section on page 7-7.
Step 5	**action** *action-statement*	Configures an action statement for the policy. See the "Configuring Action Statements" section on page 7-8. Repeat Step 5 for multiple action statements.

	Command	Purpose
Step 6	**show event manager policy internal** *name*	(Optional) Displays information about the configured policy.
Step 7	**copy running-config startup-config**	(Optional) Saves this configuration change.

Configuring Event Statements

To configure an event statement, use one the following commands in EEM configuration mode:

Command	Purpose
event cli match *expression* [**count** repeats | **time** *seconds*]	Triggers an event if you enter a CLI command that matches the regular expression. The *repeats* range is from 1 to 65000. The time range, in seconds, is from 0 to 4294967295.
event counter name *counter* **entry-val** *entry* **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** |**ne**} [**exit-val** *exit* **exit-op** {**eq** | **ge** | **gt** | **le** | **lt** |**ne**}]	Triggers an event if the counter crosses the entry threshold (based on the entry operation—greater than, less than, and so on.) The event resets immediately. Optionally, you can configure the event to reset after the counter passes the exit threshold. The *counter* name can be any case-sensitive, alphanumeric string up to 28 characters. The *entry* and *exit* value ranges are from 0 to 2147483647.
event fanabsent [**fan** *number*] **time** *seconds*	Triggers an event if a fan is removed from the device for more than the configured time, in seconds. The fan *number* range is dependent on different switches (for example for 9513 switches the range is from1 to 2, for 9506/9509 switches the range is 1). The *seconds* range is from 10 to 64000.
event fanbad [**fan** *number*] **time** *seconds*	Triggers an event if a fan fails for more than the configured time, in seconds. The fan *number* range is dependent on different switches (for example for 9513 switches the range is from1 to 2, for 9506/9509 switches the range is 1). The *seconds* range is from 10 to 64000.
event memory {**critical** | **minor** | **severe**}	Triggers an event if a memory threshold is crossed.
event module-failure type *failure-type* ***module*** {*slot* | **all**} **count** *repeats* [**time** *seconds*]	Triggers an event if a module experiences the failure type configured. The *slot* range is dependent on different switches (for example for 9513 switches the range is from1 to 13, for 9509 switches the range is 1 to 9). The *repeats* range is from 0 to 4294967295. The *seconds* range is from 0 to 4294967295.

Command	Purpose
event oir {**fan** | **module** | **powersupply**} {**anyoir** | **insert** | **remove**} [*number*]	Triggers an event if the configured device element (fan, module, or power supply) is inserted or removed from the device. You can optionally configure a specific fan, module, or power supply number. The *number* range is as follows: • Fan number is dependent on different switches. • Module number is dependent on different switches. • Power supply number range is from 1 to 2.
event policy-default count *repeats* [**time** *seconds*]	Uses the event configured in the system policy. Use this option for overriding policies. The *repeats* range is from 1 to 65000. The *seconds* range is from 0 to 4294967295.
event poweroverbudget	Triggers an event if the power budget exceeds the capacity of the configured power supplies.
event snmp oid *oid* **get-type** {**exact** | **next**} **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** |**ne**} **entry-val** *entry* [**exit-comb** {**and** | **or**}] **exit-op** {**eq** | **ge** | **gt** | **le** | **lt** |**ne**} **exit-val** *exit* **exit-time** *time* **polling-interval** *interval*	Triggers an event if the SNMP OID crosses the entry threshold (based on the entry operation—greater than, less than, and so on.) The event resets immediately, or optionally you can configure the event to reset after the counter passes the exit threshold. The OID is in dotted decimal notation. The *entry* and *exit* value ranges are from 0 to 18446744073709551615. The time range is from 0 to 2147483647. The interval range is from 1 to 2147483647.
event temperature [**module** *slot*] [*sensor number*] **threshold** {**any** | **major** | **minor**}	Triggers an event if the temperature sensor exceeds the configured threshold. The *slot* range is dependent on different switches. The *sensor range* is from 1 to 8 on MDS modules, but current MDS modules use the range from 1 to 3 only, some modules use the range from 1 to 2.

Configuring Action Statements

To configure action statements, use the following commands in EEM configuration mode:

Command	Purpose
action *number*[.*number2*] **cli** *command1* [command2...] [**local**]	Executes the configured CLI commands. You can optionally execute the commands on the module where the event occurred. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[.*number2*] **counter name** *counter* **value** *val* **op** {**dec** | **inc** | **nop** | **set**}	Modifies the counter by the configured value and operation. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The counter name can be any case-sensitive, alphanumeric string up to 28 characters. The *val* can be an integer from 0 to 2147483647 or a substituted parameter.
action *number*[.*number2*] **event-default**	Executes the default action for the associated event. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action number [.number2] **exceptionlog module** module **syserr** error **devid** id **errtype** type **errcode** code **phylayer** layer **ports** list **harderror** error [**desc** string]	Logs an exception if the specific conditions are encountered when an EEM applet is triggered.
action *number*[.*number2*] **forceshut** [**module** *slot* | **xbar** xbar-*number*] **reset-reason** *seconds*	Forces a module, crossbar, or the entire system to shut down. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *slot* range is dependent on different switches. The *xbar-number* range is from 1 to 2 and is only available on MDS 9513 modules. The reset reason is a quoted alphanumeric string up to 80 characters.
action *number*[.*number2*] **overbudgetshut** [**module** *slot* [**-** *slot*]]	Forces one or more modules or the entire system to shut down because of a power overbudget issue. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *slot* range is dependent on different switches.
action *number*[.*number2*] **policy-default**	Executes the default action for the policy that you are overriding. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.

Command	Purpose
action *number*[.*number2*] **reload** [**module** *slot* [**-** *slot*]]	Forces one or more modules or the entire system to reload. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *slot* range is dependent on different switches.
action *number*[.*number2*] **snmp-trap** {[**intdata1** *data* [**intdata2** *data*] [strdata string]}	Sends an SNMP trap with the configured data. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *data* arguments can by any number up to 80 digits. The *string* can be any alphanumeric string up to 80 characters.
action *number*[.*number2*] **syslog** [**priority** *prio-val*] **msg** *error-message*	Sends a customized syslog message at the configured priority.*number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *error-message* can be any quoted alphanumeric string up to 80 characters.

Note If you want to allow the triggered event to process the default actions also, you must explicitly configure an EEM action with **event-default** or **policy-default**, based on the type of policy. For example, if you match a CLI command in a match statement, you must add the **event-default** action statement to the EEM policy or EEM will not allow the CLI command to execute. You can bypass all CLI based EEM policies using **terminal event-manager bypass** command. To revert use **terminal no event-manager bypass** command.

Defining a Policy Using a VSH Script

To define a policy using a VSH script, follow these steps:

Step 1 In a text editor, list the CLI commands that define the policy.

Step 2 Name the text file and save it.

Step 3 Copy the file to the following system directory:

bootflash://eem/user_script_policies

Registering and Activating a VSH Script Policy

To register and activate a policy defined in a VSH script, follow these steps:

	Command	Purpose
Step 1	**config t**	Enters configuration mode.
Step 2	**event manager policy** *policy-script*	Registers and activates an EEM script policy. The *policy-script* can be any case-sensitive alphanumeric string up to 29 characters.
Step 3	**show event manager internal policy** *name*	(Optional) Displays information about the configured policy.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

Overriding a Policy

To override a system policy, follow these steps:

	Command	Purpose
Step 1	**config t**	Enters configuration mode.
Step 2	**show event manager policy-state** *system-policy*	(Optional) Displays information about the system policy that you want to override, including thresholds. Use the **show event manager system-policy** command to find the system policy names.
Step 3	**event manager applet** *applet-name* **override** *system-policy*	Overrides a system policy and enters applet configuration mode. The *applet-name* can be any case-sensitive alphanumeric string up to 29 characters. The *system-policy* must be one of the existing system policies.
Step 4	**description** *policy-description*	(Optional) Configures a descriptive string for the policy. The string can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks.
Step 5	**event** *event-statement*	Configures the event statement for the policy. See the "Configuring Event Statements" section on page 7-7.
Step 6	**action** *action-statement*	Configures an action statement for the policy. See the "Configuring Action Statements" section on page 7-8. Repeat Step 6 for multiple action statements.
Step 7	**show event manager policy-state** *name*	(Optional) Displays information about the configured policy.
Step 8	**copy running-config startup-config**	(Optional) Saves this configuration change.

Defining an Environment Variable

To define a variable to serve as a parameter in an EEM policy, follow these steps:

	Command	Purpose
Step 1	**config t**	Enters configuration mode.
Step 2	**event manager environment** *variable-name variable-value*	Creates an environment variable for EEM. The *variable-name* can be any case-sensitive alphanumeric string up to 29 characters. The *variable-value* can be any quoted alphanumeric string up to 39 characters.
Step 3	**show event manager environment**	(Optional) Displays information about the configured environment variables.
Step 4	**copy running-config startup-config**	(Optional) Saves this configuration change.

Verifying EEM Configuration

To display EEM configuration information, perform one of the following tasks:

Command	Purpose
show event manager environment [*variable-name* | **all**]	Displays information about the event manager environment variables.
show event manager event-types [*event* | **all** | **module** *slot*]	Displays information about the event manager event types.
show event manager history events [**detail**] [**maximum** *num-events*] [**severity** {**catastrophic** | **minor** | **moderate** | **severe**}]	Displays the history of events for all policies.
show event manager policy internal [*policy-name*] [**inactive**]	Displays information about the configured policies.
show event manager policy-state *policy-name*	Displays information about policy state, including thresholds.
show event manager script system [*policy-name* | **all**]	Displays information about the script policies.
show event manager system-policy [**all**]	Displays information about the predefined system policies.
show running-config eem	Displays information about the running configuration for EEM.
show startup-config eem	Displays information about the startup configuration for EEM.

EEM Example Configuration

This example overrides the __lcm_module_failure system policy by changing the threshold for just module 3 hitless upgrade failures. This example also sends a syslog message. The settings in the system policy, __lcm_module_failure, apply in all other cases.

```
event manager applet example2 override __lcm_module_failure
 event module-failure type hitless-upgrade-failure module 3 count 2
 action 1 syslog priority errors msg module 3 "upgrade is not a hitless upgrade!"
 action 2 policy-default
```

This example creates an EEM policy that allows the CLI command to execute but triggers an SNMP notification when a user enters configuration mode on the device:

```
event manager applet TEST
event cli match "conf t"
action 1.0 snmp-trap strdata "Confiiguration change"
action 2.0 event-default
```

Note You must add the **event-default** action statement to the EEM policy or EEM will not allow the CLI command to execute.

Default Settings

Table 7-1 lists the default settings for EEM parameters.

Table 7-1 Default EEM Parameters

Parameters	Default
system policies	active