



Configuring EEM

This chapter contains the following sections:

- [Information About Embedded Event Manager, page 1](#)
- [EEM Policies, page 2](#)
- [EEM Event Statement, page 3](#)
- [EEM Action Statements, page 4](#)
- [VSH Script Policies, page 4](#)
- [EEM Event Correlation, page 4](#)
- [EEM Virtualization Support, page 5](#)
- [EEM Licensing Requirements, page 5](#)
- [Prerequisites for EEM, page 5](#)
- [Guidelines and Limitations for EEM, page 5](#)
- [Default Settings for EEM, page 6](#)
- [Configuring EEM, page 6](#)
- [Verifying the EEM Configuration, page 17](#)
- [Configuration Examples for EEM, page 18](#)

Information About Embedded Event Manager

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

EEM consists of three major components:

- Event statements—Events to monitor from another Cisco NX-OS component that might 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.

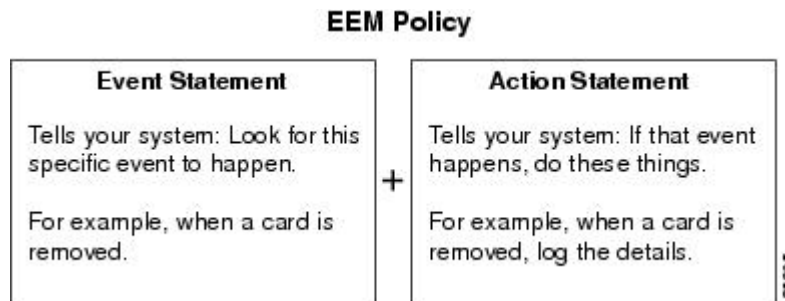
- Policies—An event paired with one or more actions to troubleshoot or recover from the event.

EEM 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.

The following figure shows the two basic statements in an EEM policy.

Figure 1: EEM Policy Statement



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

EEM gives you a device-wide view of policy management. You configure EEM policies on the supervisor, and EEM pushes the policy to the correct module based on the event type. EEM takes any actions for a triggered event either locally on the module or on the supervisor (the default option).

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 that are related to the same event as your policy. To configure a user policy, see [Defining a User Policy Using the CLI, on page 7](#).

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 [Overriding a Policy, on page 12](#).



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.

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

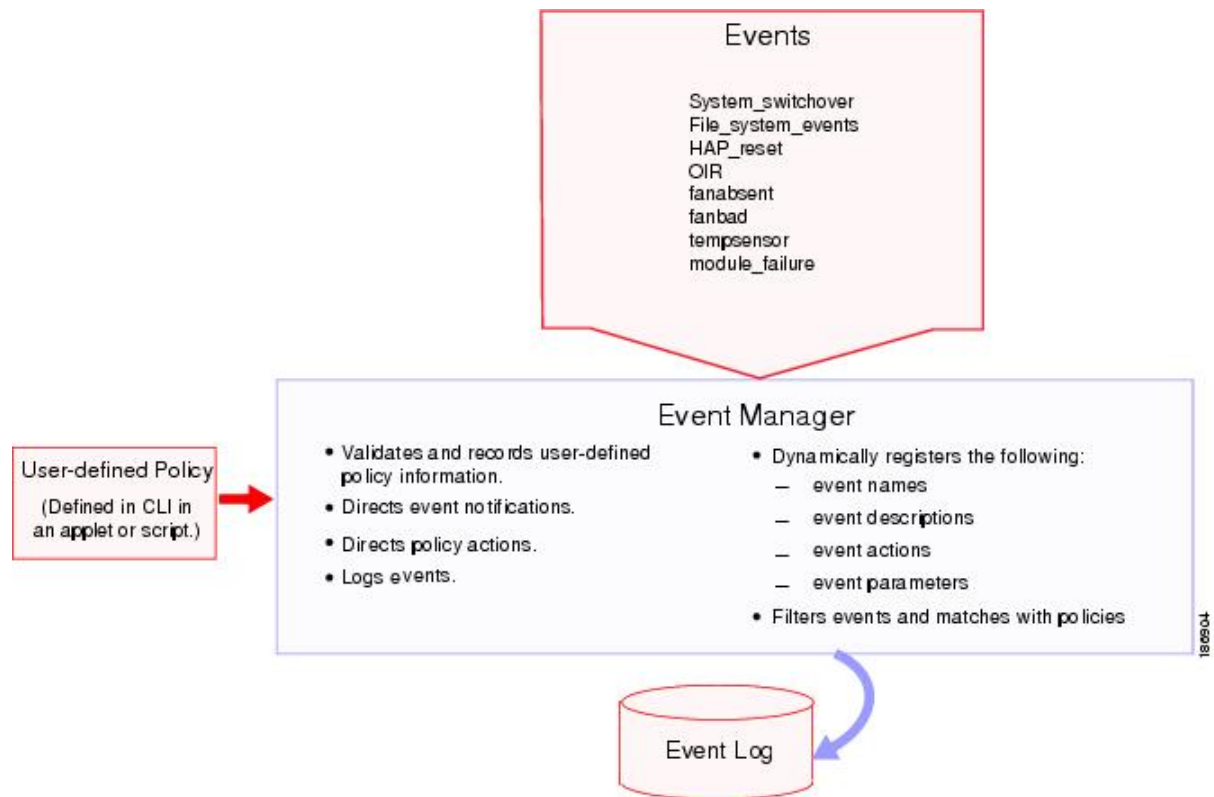
EEM Event Statement

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 that only critical events or multiple occurrences of an event within a specified time period trigger an associated action.

The following figure shows events that are handled by EEM.

Figure 2: EEM Overview



Event statements specify the event that triggers a policy to run. You can configure multiple event triggers. For more information on configuring multiple events, see [EEM Event Correlation, on page 4](#).

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 any default actions, you must configure the EEM policy to allow the event default action statement.

EEM Action Statements

Action statements describe the action that is 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.
- Reload the device.
- Generate a syslog message.
- Generate an SNMP notification.
- Use the default action for the system policy.

**Note**

If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the default action. 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 does not allow the CLI command to execute.

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 policies. After you write your VSH script policy, copy it to the device and activate it. To configure a policy in a VSH script, see [Defining a Policy Using a VSH Script](#), on page 11.

EEM Event Correlation

You can trigger an EEM policy based on a combination of events. First, you use the **tag** keyword to create and differentiate multiple events in the EEM policy. Then, using a set of Boolean operators (and, or, andnot), with the count and time, you can define a combination of these events to trigger a custom action.

**Note**

For information about configuring EEM event correlation, see [Defining a User Policy Using the CLI](#), on page 7.

EEM Virtualization Support

You configure EEM in the virtual device context (VDC) that you are logged into. By default, Cisco NX-OS places you in the default VDC. You must be in this VDC to configure policies for module-based events.

Not all actions or events are visible in all VDCs. You must have network-admin or vdc-admin privileges to configure policies.

EEM Licensing Requirements

| Product | License Requirement |
|-------------|--|
| Cisco 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. For a complete explanation of the Cisco NX-OS licensing scheme, see the Cisco NX-OS Licensing Guide. |

Prerequisites for EEM

EEM has the following prerequisite:

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

Guidelines and Limitations for EEM

EEM has the following configuration guidelines and limitations:

- The maximum number of configurable EEM policies is 500.
- 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 a triggered event to process any default actions, you must configure the EEM policy to allow the default action. 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 does 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.
- In regular command expressions, all keywords must be expanded and only the * symbol can be used for argument replacement.
- EEM event correlation supports up to four event statements in a single policy. The event types can be the same or different, but only these event types are supported: cli, counter, module, module-failure, oir, snmp, syslog, and track.

- When more than one event statement is included in an EEM policy, each event statement must have a **tag** keyword with a unique *tag* argument.
- EEM event correlation does not override the system default policies.
- Default action execution is not supported for policies that are configured with tagged events.

Default Settings for EEM

The following table lists the default setting for EEM parameters:

Table 1: Default EEM Parameters

| Parameters | Default |
|-----------------|---------|
| System Policies | Active |

Configuring EEM

Defining an Environment Variable

You can define a variable to serve as a parameter in an EEM policy.

Before You Begin

Make sure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# event manager environment <i>variable-name</i> <i>variable-value</i> | Creates an environment variable for EEM. The <i>variable-name</i> can be any case-sensitive alphanumeric string up to 29 characters. The <i>variable-value</i> can be any quoted alphanumeric string up to 39 characters. |
| Step 3 | switch(config)# show event manager environment { <i>variable-name</i> all } | (Optional) Displays information about the configured environment variables. Enclose the string in quotation marks. |
| Step 4 | switch(config)# copy running-config startup-config | (Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

This example shows how to define an environment variable:

```
switch# configure terminal
switch(config)# event manager environment emailto "admin@anyplace.com"
switch(config)# show event manager environment all
switch(config)# copy running-config startup-config
```

Defining a User Policy Using the CLI

You can define a user policy using the CLI.

Before You Begin

Make sure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# event manager applet <i>applet-name</i> | Registers the applet with EEM and enters applet configuration mode. The <i>applet-name</i> can be any case-sensitive, alphanumeric string up to 29 characters. |
| Step 3 | switch(config-applet)# description <i>policy-description</i> | (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 | switch(config-applet)# event <i>event-statement</i> | Configures the event statement for the policy. See Event Statement Configuration , on page 8. Repeat Step 4 for multiple event statements. |
| Step 5 | switch(config-applet)# tag <i>tag</i> { and andnot or } <i>tag</i> [and andnot or { <i>tag</i> }] { happens occurs in seconds } | (Optional) Correlates multiple events in the policy. The range for the <i>occurs</i> argument is from 1 to 4294967295. The range for the <i>seconds</i> argument is from 0 to 4294967295 seconds. |
| Step 6 | switch(config-applet)# action <i>number</i> [<i>number2</i>] <i>action-statement</i> | Configures an action statement for the policy. See Action Statement Configuration , on page 10. Repeat Step 6 for multiple action statements. |
| Step 7 | switch(config-applet)# show event manager policy-state <i>name</i> [<i>module module-id</i>] | Displays information about the status of the configured policy. |
| Step 8 | switch(config-applet)# copy running-config startup-config | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

This example shows how to define a user policy by using the CLI:

```
switch# configure terminal
switch(config)# event manager applet monitorShutdown
switch(config-applet)# description "Monitors interface shutdown."
switch(config-applet)# event cli match "shutdown"
switch(config-applet)# tag one or two happens 1 in 10000
switch(config-applet)# action 1.0 cli local show interface e 3/1
switch(config-applet)# show event manager policy-state monitorShutdown
switch(config-applet)# copy running-config startup-config
```

Event Statement Configuration

Use one of the following commands in EEM configuration mode to configure an event statement:

| Command | Purpose |
|--|---|
| event cli [tag tag] match <i>expression</i> [count repeats time seconds] Example: switch(config-applet)# event cli match "shutdown" | Triggers an event if you enter a command that matches the regular expression. The tag tag keyword-argument pair identifies this specific event when multiple events are included in the policy. The <i>repeats</i> argument range is from 1 to 65000. The <i>seconds</i> argument range is from 0 to 4294967295, where 0 indicates no time limit. Enclose the string in quotation marks. |
| event counter [tag tag] name <i>counter</i> entry-val <i>entry</i> entry-op { eq ge gt le lt ne } [exit-val <i>exit</i> exit-op { eq ge gt le lt ne }] Example: switch(config-applet)# event counter name mycounter entry-val 20 gt | Triggers an event if the counter crosses the entry threshold based on the entry operation. The event resets immediately. Optionally, you can configure the event to reset after the counter passes the exit threshold. The tag tag keyword-argument pair identifies this specific event when multiple events are included in the policy. The <i>counter</i> name can be any case-sensitive, alphanumeric string up to 28 characters. The <i>entry</i> and <i>exit</i> value ranges are from 0 to 2147483647. |
| event fanabsent [fan number] time <i>seconds</i> Example: switch(config-applet)# event fanabsent time 3000 | Triggers an event if a fan is removed from the device for more than the configured time, in seconds. The <i>number</i> range is module dependent. The <i>seconds</i> range is from 10 to 64000. |
| event fanbad [fan number] time <i>seconds</i> Example: switch(config-applet)# event fanbad time 3000 | Triggers an event if a fan fails for more than the configured time, in seconds. The <i>number</i> range is module dependent. The <i>seconds</i> range is from 10 to 64000. |

| Command | Purpose |
|--|--|
| event oir [tag tag] { fan module powersupply } { anyoir insert remove } [<i>number</i>] Example: switch(config-applet)# event oir fan remove 4 | <p>Triggers an event if the configured device element (fan, module, or power supply) is inserted or removed from the device.</p> <p>The tag tag keyword-argument pair identifies this specific event when multiple events are included in the policy.</p> <p>You can optionally configure a specific fan, module, or power supply number. The <i>number</i> range is as follows:</p> <ul style="list-style-type: none"> • Fan number—Module dependent. • Module number—Device dependent. • Power supply number—The range is from 1 to 3. |
| event policy-default count <i>repeats</i> [<i>time seconds</i>] Example: switch(config-applet)# event policy-default count 3 | <p>Uses the event configured in the system policy. Use this option for overriding policies.</p> <p>The <i>repeats</i> range is from 1 to 65000. The <i>seconds</i> range is from 0 to 4294967295, where 0 indicates no time limit.</p> |
| event snmp [tag tag] 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 Example: switch(config-applet)# event snmp oid 1.3.6.1.2.1.31.1.1.1.6 get-type next entry-op lt 300 entry-val 0 exit-op eq 400 exit-time 30 polling-interval 300 | <p>Triggers an event if the SNMP OID crosses the entry threshold based on the entry operation. The event resets immediately. Optionally, you can configure the event to reset after the counter passes the exit threshold. The OID is in dotted decimal notation.</p> <p>The tag tag keyword-argument pair identifies this specific event when multiple events are included in the policy.</p> <p>The <i>entry</i> and <i>exit</i> value ranges are from 0 to 18446744073709551615. The time, in seconds, is from 0 to 2147483647. The interval, in seconds, is from 1 to 2147483647.</p> |
| event storm-control Example: switch(config-applet)# event storm-control | <p>Triggers an event if the traffic on a port exceeds the configured storm control threshold.</p> |
| event sysmgr memory [module module-num] major major-percent minor minor-percent clear clear-percent Example: switch(config-applet)# event sysmgr memory minor 80 | <p>Triggers an event if the specified system manager memory threshold is exceeded. The range for the percentage is from 1 to 99.</p> |

| Command | Purpose |
|--|---|
| event temperature [<i>module slot</i>] [<i>sensor number</i>] threshold { <i>any</i> <i>major</i> <i>minor</i> } Example: switch(config-applet)# event temperature module 2 threshold any | Triggers an event if the temperature sensor exceeds the configured threshold. The sensor range is from 1 to 18. |
| event track [<i>tag tag</i>] <i>object-number</i> state { <i>any</i> <i>down</i> <i>up</i> } Example: switch(config-applet)# event track 1 state down | Triggers an event if the tracked object is in the configured state. The tag tag keyword-argument pair identifies this specific event when multiple events are included in the policy. The <i>object-number</i> range is from 1 to 500. |

Action Statement Configuration

Use the following commands in EEM configuration mode to configure action statements:

| Command | Purpose |
|--|---|
| action <i>number</i> [<i>number2</i>] cli <i>command1</i> [<i>command2...</i>] [<i>local</i>] Example: switch(config-applet)# action 1.0 cli "show interface e 3/1" | Runs the configured CLI commands. You can optionally run the commands on the module where the event occurred. The action label is in the format <i>number1.number2</i> . <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. Enclose the string in quotation marks. |
| action <i>number</i> [<i>number2</i>] counter name <i>counter value val</i> op { <i>dec</i> <i>inc</i> <i>nop</i> <i>set</i> } Example: switch(config-applet)# action 2.0 counter name mycounter value 20 op inc | Modifies the counter by the configured value and operation. The action label is in the format <i>number1.number2</i> . <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. The counter name can be any case-sensitive, alphanumeric string up to 28 characters. The <i>val</i> can be an integer from 0 to 2147483647 or a substituted parameter. |
| action <i>number</i> [<i>number2</i>] event-default Example: switch(config-applet)# action 1.0 event-default | Executes the default action for the associated event. The action label is in the format <i>number1.number2</i> . <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. |

| Command | Purpose |
|---|--|
| action <i>number</i> [. <i>number2</i>] policy-default Example: switch(config-applet)# action 1.0 policy-default | Executes the default action for the policy that you are overriding. The action label is in the format <i>number1.number2</i> . <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. |
| action <i>number</i> [. <i>number2</i>] reload [module <i>slot</i> [- <i>slot</i>]] Example: switch(config-applet)# action 1.0 reload module 3-5 | Forces one or more modules or the entire system to reload. <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. |
| action <i>number</i> [. <i>number2</i>] snmp-trap {[intdata1 <i>data</i> [intdata2 <i>data</i>] [strdata <i>string</i>]} Example: switch(config-applet)# action 1.0 snmp-trap strdata "temperature problem" | Sends an SNMP trap with the configured data. <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. The <i>data</i> arguments can be any number up to 80 digits. The <i>string</i> can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks. |
| action <i>number</i> [. <i>number2</i>] syslog [priority <i>prio-val</i>] msg <i>error-message</i> Example: switch(config-applet)# action 1.0 syslog priority notifications msg "cpu high" | Sends a customized syslog message at the configured priority. <i>number</i> can be any number up to 16 digits. The range for <i>number2</i> is from 0 to 9. The <i>error-message</i> can be any quoted alphanumeric string up to 80 characters. |

**Note**

If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the default action. 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 does not allow the CLI command to execute. You can use the **terminal event-manager bypass** command to allow all EEM policies with CLI matches to execute the CLI command.

Defining a Policy Using a VSH Script

You can define a policy using a VSH script.

Before You Begin

Make sure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Ensure that you are logged in with administrator privileges.

Ensure that your script name is the same name as the script filename.

Procedure

-
- Step 1** In a text editor, list the 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

You can register and activate a policy defined in a VSH script.

Before You Begin

Make sure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# event manager policy <i>policy-script</i> | Registers and activates an EEM script policy. The <i>policy-script</i> can be any case-sensitive, alphanumeric string up to 29 characters. |
| Step 3 | switch(config)# show event manager policy internal <i>name</i> | (Optional) Displays information about the configured policy. |
| Step 4 | switch(config)# copy running-config startup-config | (Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

This example shows how to register and activate a VSH script policy:

```
switch# configure terminal
switch(config)# event manager policy moduleScript
switch(config)# show event manager policy internal moduleScript
switch(config)# copy running-config startup-config
```

Overriding a Policy

You can override a system policy.

Before You Begin

Make sure that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# show event manager policy-state <i>system-policy</i> | 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 | switch(config)# event manager applet <i>applet-name</i> override <i>system-policy</i> | Overrides a system policy and enters applet configuration mode. The <i>applet-name</i> can be any case-sensitive, alphanumeric string up to 29 characters. The <i>system-policy</i> must be one of the existing system policies. |
| Step 4 | switch(config-applet)# description <i>policy-description</i> | (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 | switch(config-applet)# event <i>event-statement</i> | Configures the event statement for the policy. See the “Configuring Event Statements” section on page 14-10. |
| Step 6 | switch(config-applet)# action <i>number</i> <i>action-statement</i> | Configures an action statement for the policy. See the “Configuring Action Statements” section on page 14-13. Repeat Step 6 for multiple action statements. |
| Step 7 | switch(config-applet)# show event manager policy-state <i>name</i> | (Optional) Displays information about the configured policy. |
| Step 8 | switch(config-applet)# copy running-config startup-config | Saves this configuration change. |

This example shows how to override a policy:

```
switch# configure terminal
switch(config)# show event manager policy-state _ethpm_link_flap
Policy _ethpm_link_flap
  cfg count : 5
  cfg time interval : 10.000000 (seconds)
  Hash default, Count 0
switch(config)# event manager applet ethport override _ethpm_link_flap
switch(config-applet)# description "Overrides link flap policy"
switch(config-applet)# event policy-default count 2 time 1000
switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."
switch(config-applet)# show event manager policy-state ethport
switch(config-applet)# copy running-config startup-config
```

Configuring the Syslog as an EEM Publisher

You can monitor syslog messages from the switch.

**Note**

The maximum number of searchable strings to monitor syslog messages is 10.

Before You Begin

Make sure that the EEM is available for registration by syslog.

Configure and execute the syslog daemon.

Procedure

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | <code>switch# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>switch(config)# event manager applet <i>applet-name</i></code> | Registers an applet with EEM and enters applet configuration mode. |
| Step 3 | <code>switch(config)# event syslog [tag <i>tag</i>] {occurs <i>number</i> period <i>seconds</i> pattern <i>msg-text</i> priority <i>priority</i>}</code> | Monitors syslog messages and invokes the policy based on the search string in the policy. <ul style="list-style-type: none"> • The tag <i>tag</i> keyword-argument pair identifies this specific event when multiple events are included in the policy. • The occurs <i>number</i> keyword-argument pair specifies the number of occurrences. The range is from 1 to 65000. • The period <i>seconds</i> keyword-argument pair specifies the interval during which the event occurs. The range is from 1 to 4294967295. • The pattern <i>msg-text</i> keyword-argument pair specifies the matching regular expression. The pattern can contain character text, an environment variable, or a combination of the two. If the string contains embedded blanks, it is enclosed in quotation marks. • The priority <i>priority</i> keyword-argument pair specifies the priority of the syslog messages. If this keyword is not selected, all syslog messages are set at the informational priority level. |
| Step 4 | <code>switch(config)# copy running-config startup-config</code> | (Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

This example shows how to configure the syslog as an EEM publisher:

```
switch# configure terminal
switch(config)# event manager applet abc
switch(config-applet)# event syslog occurs 10
switch(config-applet)# copy running-config startup-config
```

Defining a User Policy Using the CLI to Trigger a Tcl Script

Before You Begin

Copy the Tcl script which is triggered through an EEM policy to the bootflash of the switch.

Procedure

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# event manager applet <i>applet-name</i> | Registers the applet with EEM and enters applet configuration mode. The <i>applet-name</i> can be any case-sensitive, alphanumeric string up to 29 characters. |
| Step 3 | switch(config-applet)# description <i>policy-description</i> | (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 | switch(config-applet)# event <i>event-statement</i> | Configures the event statement for the policy. See Event Statement Configuration , on page 8. Repeat Step 4 for multiple event statements. |
| Step 5 | switch(config-applet)# tag <i>tag</i> { and andnot or } <i>tag</i> [and andnot or { <i>tag</i> }] { happens occurs in seconds } | (Optional) Correlates multiple events in the policy. The range for the <i>occurs</i> argument is from 1 to 4294967295. The range for the <i>seconds</i> argument is from 0 to 4294967295 seconds. |
| Step 6 | switch(config-applet)# action <i>number</i> [<i>number2</i>] <i>action-statement</i> tcl- <i>filename</i> | Configures an action statement for the policy. See Action Statement Configuration , on page 10. Repeat Step 6 for multiple action statements. |
| Step 7 | switch(config-applet)# show event manager policy-state <i>name</i> [module <i>module-id</i>] | Displays information about the status of the configured policy. |
| Step 8 | switch(config-applet)# copy running-config startup-config | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

Sample Tcl file (Vlan.tcl). Copy this file to the bootflash. Running the file creates 99 VLANs and names them.

```
set i 1
while {$i<100} {
cli configure terminal
cli vlan $i
cli name VLAN$i
cli no shutdown
}
```

```
cli exit
incr i
}
```

This example shows how to define a user policy by using the CLI and invoking a Tcl script by using the action statement once the event is triggered:

```
switch# configure terminal
switch(config)# event manager applet TCL
switch(config-applet)# description "Triggers TCL Script"
switch(config-applet)# event cli match "shutdown"
switch(config-applet)# action 1.0 cli local tclsh VLAN.tcl
switch(config-applet)# copy running-config startup-config
```

Defining a User Policy Using the CLI to Trigger a Python Script

Before You Begin

Copy the Python script which is triggered through an EEM policy to the bootflash of the switch.

Procedure

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | switch# configure terminal | Enters global configuration mode. |
| Step 2 | switch(config)# event manager applet <i>applet-name</i> | Registers the applet with EEM and enters applet configuration mode. The <i>applet-name</i> can be any case-sensitive, alphanumeric string up to 29 characters. |
| Step 3 | switch(config-applet)# description <i>policy-description</i> | (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 | switch(config-applet)# event <i>event-statement</i> | Configures the event statement for the policy. See Event Statement Configuration , on page 8. Repeat Step 4 for multiple event statements. |
| Step 5 | switch(config-applet)# tag <i>tag</i> { and andnot or } <i>tag</i> [and andnot or { <i>tag</i> }] { happens <i>occurs in seconds</i> } | (Optional) Correlates multiple events in the policy. The range for the <i>occurs</i> argument is from 1 to 4294967295. The range for the <i>seconds</i> argument is from 0 to 4294967295 seconds. |
| Step 6 | switch(config-applet)# action <i>number</i> [. <i>number2</i>] <i>action-statementpython-filename</i> | Configures an action statement for the policy. See Action Statement Configuration , on page 10. Repeat Step 6 for multiple action statements. |
| Step 7 | switch(config-applet)# show event manager policy-state <i>name</i> [module <i>module-id</i>] | Displays information about the status of the configured policy. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 8 | <code>switch(config-applet)# copy running-config startup-config</code> | Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration. |

Sample Python file (Python.py). Copy this file to the bootflash.

```
import re
import cisco
cisco.cli ("show interface eth 1/1-32 transceiver detail >> bootflash:link_flap.txt")
```

This example shows how to define a user policy by using the CLI and invoking a Python script by using the action statement once the event is triggered:

```
switch# configure terminal
switch(config)# event manager applet PYTHON
switch(config-applet)# description "Triggers PYTHON Script"
switch(config-applet)# event cli match "shutdown"
switch(config-applet)# action 1.0 cli local python bootflash:Python.py
switch(config-applet)# copy running-config startup-config
```

Verifying the EEM Configuration

To display EEM configuration information, enter one of the following commands:

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

| Command | Purpose |
|--------------------------------|---|
| show running-config eem | Displays information about the running configuration for EEM. |
| show startup-config eem | Displays information about the startup configuration for EEM. |

Configuration Examples for EEM

This example shows how to override the `__ethpm_link_flap` system policy and shut down the interface:

```
switch# configure terminal
switch(config)# event manager applet ethport override __ethpm_link_flap
switch(config-applet)# event policy-default count 2 time 1000
switch(config-applet)# action 1 cli conf t
switch(config-applet)# action 2 cli int et1/1
switch(config-applet)# action 3 cli no shut
```

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

```
switch# configure terminal
switch(config)# event manager applet TEST
switch(config-applet)# event cli match "conf t"
switch(config-applet)# action 1.0 snmp-trap strdata "Configuration change"
switch(config-applet)# action 2.0 event-default
```



Note

You must add the event-default action statement to the EEM policy, or EEM does not allow the CLI command to execute.

This example shows how to correlate multiple events in an EEM policy and execute the policy based on a combination of the event triggers. In this example, the EEM policy is triggered if one of the specified syslog patterns occurs within 120 seconds:

```
switch# configure terminal
switch(config)# event manager applet eem-correlate
switch(config-applet)# event syslog tag one pattern "copy bootflash:.* running-config.*"
switch(config-applet)# event syslog tag two pattern "copy run start"
switch(config-applet)# event syslog tag three pattern "hello"
switch(config-applet)# tag one or two or three happens 1 in 120
switch(config-applet)# action 1.0 reload module 1
```