

EEM Configuration for Cisco Integrated Services Router Platforms

Last updated: February 2008

Abstract

This document describes how to configure Cisco IOS® Embedded Event Manager (EEM) to match the scenarios in which Cisco® integrated-services-router platforms will be deployed. It also describes how to program EEM Tool Command Language (TCL) policies.

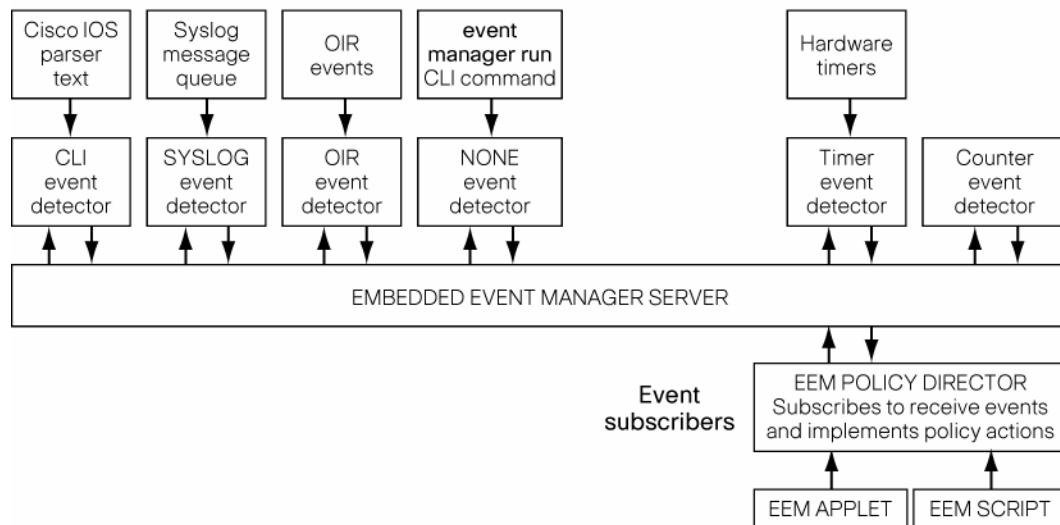
Introduction

Cisco IOS Embedded Event Manager (EEM) is a powerful tool integrated with Cisco IOS Software for system management from within the device itself. EEM offers the ability to monitor events and take informational, corrective, or any desired action when the monitored events occur or when a threshold is reached. 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 consists of three main components: EEM server, event publisher (event detector; Figure 1), and event Subscriber (policies).

Figure 1. EEM Core Event Detector

Core event publishers



Event Detectors

The event detector notifies the EEM server when an event of interest occurs. 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 (refer to Table 1).

Table 1. Event Detector

Event Detector	Description
Application-Specific Event Detector	Allows any EEM policy to publish an event
Command-Line Interface (CLI) Event Detector	Triggers policies based on commands entered through the CLI; uses a regular expression match
Counter Event Detector	Triggers policies based on a change of the designated counter; is used to manipulate counters named by the policy writer and is internal to EEM
Enhanced-Object-Tracking Event Detector*	Triggers policies when the status of tracked object changes
Interface-Counter Event Detector	Triggers policies when the Cisco IOS interface counter for a specific interface crosses a threshold
Online Insertion and Removal (OIR) Event Detector	Triggers policy when hardware is installed or removed
Resource Event Detector*	Triggers policies when Embedded Resource Manager (ERM) reports an event for specified policy; tracks resource depletion within a system
SNMP Event Detector	Triggers policies based on the associated SNMP MIB variable; includes MIB variable thresholds
Syslog Event Detector	Triggers policies based on the regular expression match of a local syslog message
Timer Event Detector	Triggers policies based on timer that includes: <ul style="list-style-type: none"> • Absolute day and time • Countdown timer down to zero • Watchdog timer • UNIX cron mechanism
None Event Detector	Triggers when the Cisco IOS event manager run CLI command executes an EEM policy
Watchdog System Monitor Event Detector	Triggers policies based on certain conditions relative to a certain Cisco IOS Software process or activity of a subsystem

* Introduced in EEM Version 2.2

EEM Policy

When an event or fault is detected, EEM policy (subscriber) implements the recovery action on the basis of the current state of the system and the actions specified in the policy for the given event. Two policy engines are defined: the Cisco IOS Software CLI applet interface and the TCL subsystem and interpreter.

The recovery action could be any of the following:

- Executing a Cisco IOS CLI command
- Generating a Cisco Networking Services (CNS) event for upstream processing by Cisco CNS devices
- Setting or modifying a named counter
- Requesting system information when an event occurs
- Sending a short e-mail
- Manually running an EEM policy
- Publishing an application-specific event
- Reloading the Cisco IOS Software
- Generating an SNMP trap

- Generating prioritized syslog messages
- Reading the state of a tracked object
- Setting the state of a tracked object

EEM Environmental Variables

EEM allows you to use environment variables in EEM policies. TCL allows you to define global variables that are known to all procedures within a TCL script. EEM allows you to define environment variables using a CLI command (**event manager environment**) for use within an EEM policy. All EEM environment variables are automatically assigned to TCL global variables before a TCL script is run. Three different types of environment variables are associated with Embedded Event Manager:

- User-defined: You can define an environment variable if you create it in a policy that you write.
- Cisco-defined: Cisco can define an environment variable for a specific sample policy.
- Cisco built-in (available in EEM applets): Cisco can define an environment variable to 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 at policy exit for policies triggered from synchronous events.

EEM Applet

An EEM applet is a simple form of policy defined within the CLI configuration. In EEM applet configuration mode, three types of configuration statements are supported. The **event** commands are used to specify the event criteria to trigger the applet to run, the **action** commands are used to specify an action to perform when the EEM applet is triggered, and the **set** command is used to set the value of an EEM applet variable. Use the **show event manager policy registered** command to display a list of registered applets.

EEM Script

A script is defined off the networking device using an ASCII editor. The script is then copied to the networking device and registered with EEM. TCL scripts are supported by EEM.

EEM allows you to write and implement your own policies using TCL. Writing an EEM policy involves:

- Selecting the event for which the policy is run
- Defining the event-detector options associated with logging and responding to the event
- Choosing the actions to be followed when the event occurs

Cisco provides enhancements to TCL in the form of keyword extensions that facilitate the development of EEM policies. The main categories of keywords identify the detected event, the subsequent action, utility information, counter values, and system information. For more details about the EEM event detectors and about creating EEM policies, refer to the [Writing Embedded Event Manager Policies](#) document.

Cisco includes a set of sample policies. You can copy the sample policies to a user directory and then modify the policies, or you can write your own policies. TCL is currently the only scripting language that Cisco supports for policy creation. You can modify TCL policies by using a text

editor such as Emacs. Policies must execute within a defined number of seconds, and you can configure the time variable. The default is currently 20 seconds.

EEM Sample Configuration: EEM with Applet Policy

Tables 2 and 3 show the commands to configure the applet and the output, respectively. Then Table 4 show the command to configure EEM with TCL script

Table 2. Applet Configuration

Command	Purpose
ROUTER(config)#event manager applet ISR_CISCO	Creates and registers the applet with EEM
ROUTER(config-applet)# event syslog pattern "Interface GigabitEthernet0/0, changed state to down"	Configures syslog event detector to match the interface message
ROUTER(config-applet)# action 1.0 cli command "enable" ROUTER(config-applet)# action 1.1 cli command "configure term" ROUTER(config-applet)# action 1.2 cli command "interface g0/1" ROUTER(config-applet)# action 1.3 cli command "no shut" ROUTER(config-applet)#	Configures action to enable the g0/1 interface on seeing the g0/0 interface going down

Table 3. Applet Output

Steps	Description
ROUTER#sh int g0/1 inc proto GigabitEthernet0/1 is administratively down, line protocol is down	The interface is manually shut down by the administrator
ROUTER(config)#int g0/0 ROUTER(config-if)#shut ROUTER(config-if)# *Nov 12 07:23:21.684: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administratively down *Nov 12 07:23:22.684: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down	Syslog message triggers the EEM and the configured action is implemented
*Nov 12 07:32:38.568: %SYS-5-CONFIG_I: Configured from console by vty0 *Nov 12 07:32:40.556: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up ROUTER(config-if)#do sh int g0/1 inc proto GigabitEthernet0/1 is up, line protocol is up ROUTER(config-if)#	

Table 4. EEM with TCL Script Policy

Command	Purpose
event manager environment <i>variable-name string</i> 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 Note: In this example, the software assigns a cron timer environment variable to be set to every second, minute, and every hour of every day.
event manager directory user policy <i>Path</i> Example: Router(config)# event manager directory user policy flash:/	Configures the location where the user-defined TCL script is stored
event manager policy <i>policy-filename [type {system user}] [trap]</i> Example: Router(config)# event manager policy tm_cli_cmd.tcl type system	Registers the EEM policy to be run when the specified event defined within the policy occurs <ul style="list-style-type: none"> Use the system keyword to register a Cisco-defined system policy. Use the user keyword to register a user-defined system policy. Note: In this example, the sample EEM policy named tm_cli_cmd.tcl is registered as a system policy.

Case Studies

The following case studies can help you understand how and where to use EEM efficiently.

Example 1: Command Execution with Logged Event

This example illustrates the use of EEM to execute show commands when a particular event occurs and collect the output and save it in some location that you can use for troubleshooting later. Figure 2 shows the topology.

Figure 2. Topology Diagram



Challenge

This example shows how to collect CPU usage and interface output when the Open Shortest Path First (OSPF) neighbor is down in router B.

Solution

EEM is configured to check for an OSPF-neighbor-down syslog message; if it occurs, it executes the following command and saves the output in flash memory:

- **show cpu process**
- **show interfaces**

The configuration follows:

```
RouterB#sh run
Building configuration...

Current configuration : 1137 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RouterB

ip cef
!

interface Loopback0
 ip address 2.2.2.2 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
```

```

interface GigabitEthernet0/1
  no ip address
  shutdown
  duplex auto
  speed auto
!
router ospf 1
  log-adjacency-changes
  network 192.168.1.0 0.0.0.255 area 0

line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  login
!
scheduler allocate 20000 1000

!
webvpn cef
!
event manager applet OSPF
  event syslog pattern "Neighbor Down: Dead timer expired"
  action 1.0 cli command "enable"
  action 1.1 cli command "sh proc cpu | append flash:cpu_info"
  action 1.2 cli command "show interface | append flash:interface_info"
  action 1.6 syslog msg "OSPF NEIGHBOR DOWN"
!
end

RouterB#
  
```

The event logs for this example follow:

```

RouterB#
RouterB#sh flas
#- --length-- -----date/time----- path
1          1902 Nov 12 2007 07:54:16 +00:00 test.tcl
3          50938004 Sep 10 2007 11:25:20 +00:00 c2800nm-advipservicesk9-
mz.124-15.T1.bin

12931072 bytes available (50946048 bytes used)

RouterB#
RouterB#
RouterB#
RouterB#sh ip ospf nei
  
```

```

Neighbor ID      Pri   State           Dead Time   Address
Interface
192.168.1.2      1    FULL/BDR       00:00:31   192.168.1.2
GigabitEthernet0/0

RouterB#sh flas
-#- --length--  -----date/time-----  path
1          1902 Nov 12 2007 07:54:16 +00:00 test.tcl
3          50938004 Sep 10 2007 11:25:20 +00:00 c2800nm-advipservicesk9-
mz.124-15.Tl.bin

12931072 bytes available (50946048 bytes used)

RouterB#
*Nov 13 07:11:26.019: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.2 on
GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired
*Nov 13 07:11:26.563: %HA_EM-6-LOG: OSPF: OSPF NEIGHBOR DOWN

RouterB#
RouterB#sh flas
-#- --length--  -----date/time-----  path
1          1902 Nov 12 2007 07:54:16 +00:00 test.tcl
3          50938004 Sep 10 2007 11:25:20 +00:00 c2800nm-advipservicesk9-
mz.124-15.Tl.bin
4          22016 Nov 13 2007 07:11:26 +00:00 cpu_info
5          3532 Nov 13 2007 07:11:26 +00:00 interface_info

12902400 bytes available (50974720 bytes used)

RouterB#

```

Example 2: Secondary MLPPP Interface Enabled when Traffic Exceeds Threshold

The Cisco integrated services router as a customer edge router plays a significant part in WAN bandwidth management. This example can help you understand the use of EEM to enable the secondary interface into Multilink Point-to-Point Protocol (MLPPP) and increase the bandwidth when the traffic exceeds the threshold. Figure 3 shows the topology.

Figure 3. Topology Diagram



Challenge

The challenge is to bring line 2 into the MLPPP bundle only when the traffic flow exceeds the configured threshold. When the traffic falls below the threshold, line 2 is unconfigured from the MLPPP bundle.

Solution

EEM is configured to check the tx_load parameter every 30 seconds, and if the parameter exceeds the configured threshold, the line 2 serial interface is configured into the MLPPP bundle. If the tx_load parameter falls below the threshold, the second line is unconfigured.

The configuration follows:

```
ISR#sh run
Building configuration...
Current configuration : 1962 bytes
!
version 12.4
hostname ISR
card type t1 0 0
!
no aaa new-model
no network-clock-participate wic 0
!
ip cef

!
voice-card 0
  no dspfarm

controller T1 0/0/0
  framing esf
  linecode b8zs
  channel-group 1 timeslots 1-24
!
controller T1 0/0/1
  framing esf
  linecode b8zs
  channel-group 1 timeslots 1-24
!
interface Multilink1
  ip address 10.1.1.2 255.255.255.0
  ppp multilink
  ppp multilink group 1
!
interface GigabitEthernet0/0
  ip address 16.16.16.1 255.255.255.0
  duplex full
  speed 100

interface Serial0/0/0:1
  no ip address
  encapsulation ppp
  ppp multilink
  ppp multilink group 1
!
interface Serial0/0/1:1
  no ip address
  encapsulation ppp
  shutdown
```

```

    ppp multilink
    !
router ospf 1
    log-adjacency-changes
    network 0.0.0.0 255.255.255.255 area 0

    !
line con 0
line aux 0
line vty 0 4
    login

event manager environment errim_period 30  ##This environment
variable specifies the frequency to check the tx_load##
event manager environment errim_int multilink1 ##This environment
variable specifies the target interface##
event manager environment sec_interface Se0/0/1:1 ##This environment
variable specifies the Second serial interface##
event manager directory user policy flash:/ ##This specifies the
location of policy TCL file##
event manager policy TX_LOAD.tcl type user ##This command register the
policy##
    !
end

ISR#

```

The event logs follow:

```

ISR#sh interface multilink 1 | inc tx
    reliability 255/255, txload 1/255, rxload 1/255
ISR#
ISR#sh ppp multilink | inc Se0/0/1:1
ISR#sh run int Se0/0/1:1
Building configuration...

Current configuration : 90 bytes
!
interface Serial0/0/1:1
    no ip address
    encapsulation ppp
    shutdown
    ppp multilink
end
ISR#
*Nov 15 04:35:30.386: %HA_EM-5-LOG:
system:/lib/tcl/eem_scripts_registered/TX_LOAD.tcl: TX Load exceeds
the threshold

```

```
*Nov 15 04:35:31.986: %HA_EM-6-LOG:
system:/lib/tcl/eem_scripts_registered/TX_LOAD.tcl: SECOND SERIAL
INTERFACE IS CONFIGURED

*Nov 15 04:35:32.030: %SYS-5-CONFIG_I: Configured from console by vty0
*Nov 15 04:35:33.210: %LINK-3-UPDOWN: Interface Serial0/0/1:1, changed
state to up
*Nov 15 04:35:34.214: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/1:1, changed state to up

ISR#sh interface multilink 1 | inc tx
      reliability 247/255, txload 14/255, rxload 14/255

ISR#sh ppp multilink | inc Se0/0/1:1
      Se0/0/1:1, since 00:00:29

ISR#sh run int se0/0/1:1
Building configuration...
Current configuration : 103 bytes
!
interface Serial0/0/1:1
  no ip address
  encapsulation ppp
  ppp multilink
  ppp multilink group 1
end

ISR#

ISR#

ISR#sh interface multilink 1 | inc tx
      reliability 232/255, txload 7/255, rxload 7/255
ISR#

*Nov 15 04:38:30.414: %HA_EM-5-LOG:
system:/lib/tcl/eem_scripts_registered/TX_LOAD.tcl: TX Load below the
threshold. So Unconfiguring the secondary interface

*Nov 15 04:38:32.014: %HA_EM-6-LOG:
system:/lib/tcl/eem_scripts_registered/TX_LOAD.tcl: SECOND SERIAL
INTERFACE IS UNCONFIGURED

*Nov 15 04:38:32.058: %SYS-5-CONFIG_I: Configured from console by vty0
*Nov 15 04:38:32.614: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/1:1, changed state to down
*Nov 15 04:38:33.618: %LINK-5-CHANGED: Interface Serial0/0/1:1,
changed state to administratively down
ISR#sh ppp multilink | inc Se0/0/1:1
ISR#sh run int s0/0/1:1
Building configuration...

Current configuration : 90 bytes
!
interface Serial0/0/1:1
```

```

no ip address
encapsulation ppp
shutdown
ppp multilink
end

```

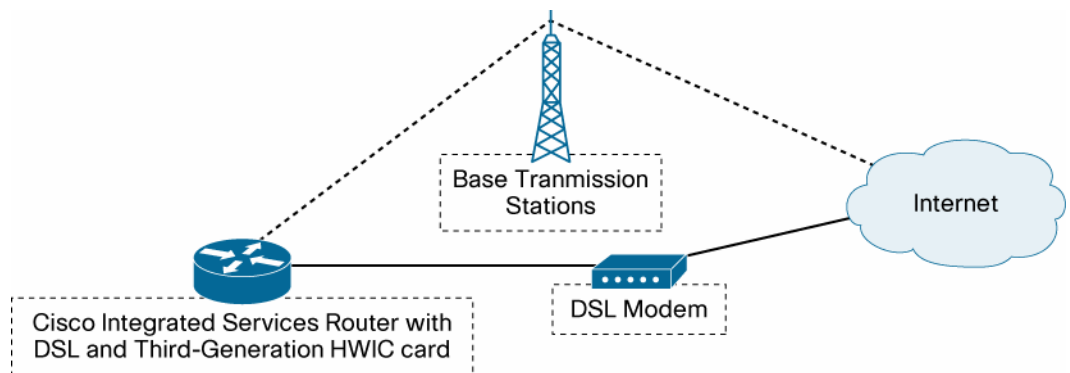
ISR#

Note: Whenever you change the TCL policy script, you need to re-register it by unconfiguring and configuring the **event manager policy <policy_name> type user** command.

Example 3: Clearing NAT Translation Table when Outgoing Interface Switches from Primary to Secondary, or Conversely

This example can help you understand how you can use EEM to clear the Network Address Translation (NAT) table when the primary link fails and the secondary link comes up, or conversely. Figure 4 shows the topology.

Figure 4. Topology Diagram



Challenge

In the topology of Figure 4, NAT is configured to translate the inside private IP address to public when it is destined to the Internet. DSL connectivity is the primary interface; if it goes down, the cellular interface will be up. When a failover occurs from the primary to the secondary interface, the NAT table will have entries for an outgoing interface that are no longer in the routing table, resulting in loss of packets.

Solution

EEM is configured to clear the NAT translation table when failover occurs.

Following is the configuration:

```

ip dhcp excluded-address 10.4.0.254
!
ip dhcp pool cdmapi
network 10.4.0.0 255.255.0.0
dns-server 66.209.10.201 66.102.163.231
default-router 10.4.0.254

chat-script cdma "" "atdt#777" TIMEOUT 30 "CONNECT"

```

```
track 234 rtr 1 reachability
crypto isakmp policy 1
  encr 3des
  authentication pre-share
crypto isakmp key abcd address 128.107.241.234
!
!
crypto ipsec transform-set abcd ah-sha-hmac esp-3des
!
crypto map cdmal 10 ipsec-isakmp
  set peer 128.107.241.234
  set transform-set abcd
  match address 103
!
!
interface ATM0/0/0
  no ip address
  ip virtual-reassembly
  load-interval 30
  no atm ilmi-keepalive
  dsl operating-mode auto
!
interface ATM0/0/0.1 point-to-point
  backup interface Cellular0/3/0
  ip nat outside
  ip virtual-reassembly
  no snmp trap link-status
  pvc 0/35
  pppoe-client dial-pool-number 2
!
!
interface Cellular0/3/0
  bandwidth receive 1400000
  ip address negotiated
  ip nat outside
  ip virtual-reassembly
  encapsulation ppp
  no ip mroute-cache
  dialer in-band
  dialer idle-timeout 0
  dialer string cdma
  dialer-group 1
  async mode interactive
  no ppp lcp fast-start
  ppp chap password 0 cisco
  ppp ipcp dns request
  crypto map cdmal
!
interface Vlan104
```

```

description used as default gateway address for DHCP clients
ip address 10.4.0.254 255.255.0.0
ip nat inside
ip virtual-reassembly

!
interface Dialer2
ip address negotiated
ip mtu 1492
ip nat outside
ip virtual-reassembly
encapsulation ppp
load-interval 30
dialer pool 2
dialer-group 2
ppp authentication chap callin
ppp chap hostname cisco@dsl.com
ppp chap password 0 cisco
ppp ipcp dns request
crypto map cdmal

!
ip local policy route-map track-primary-if
ip route 0.0.0.0 0.0.0.0 Dialer2 track 234
ip route 0.0.0.0 0.0.0.0 Cellular0/3/0 254
!
!
ip nat inside source route-map nat2cell interface Cellular0/3/0
overload
ip nat inside source route-map nat2dsl interface Dialer2 overload
!
ip sla 1
icmp-echo 209.131.36.158 source-interface Dialer2
timeout 1000
frequency 2
ip sla schedule 1 life forever start-time now
access-list 1 permit any
access-list 2 permit 10.4.0.0 0.0.255.255
access-list 3 permit any
access-list 101 permit ip 10.4.0.0 0.0.255.255 any
access-list 102 permit icmp any host 209.131.36.158
access-list 103 permit ip host 166.138.186.119 128.107.0.0 0.0.255.255
access-list 103 permit ip host 75.40.113.246 128.107.0.0 0.0.255.255
dialer-list 1 protocol ip list 1
dialer-list 2 protocol ip permit
!
!
route-map track-primary-if permit 10
match ip address 102
set interface Dialer2
!

```

```
route-map nat2dsl permit 10
  match ip address 101
  match interface Dialer2
!
route-map nat2cell permit 10
  match ip address 101
  match interface Cellular0/3/0
!
line 0/3/0
  exec-timeout 0 0
  script dialer cdma
  login
  modem InOut

event manager applet Pri_back
event track 234 state any
action 2.0 cli command "clear ip nat trans forced"
```

Writing EEM Policies

All Embedded Event Manager policies are written in TCL, a string-based command language that is interpreted at runtime. As an enforced rule, Embedded Event Manager policies are short-lived runtime routines that execute in fewer than 20 seconds, but the duration is configurable.

EEM policies use the full range of capabilities of the TCL language. However, Cisco provides enhancements to the TCL language in the form of keyword extensions that facilitate the writing of EEM policies.

EEM policy keywords identify the event or fault for which recovery actions or responsive actions are to be implemented. Keywords also register the policy with Embedded Event Manager to schedule execution of the policy.

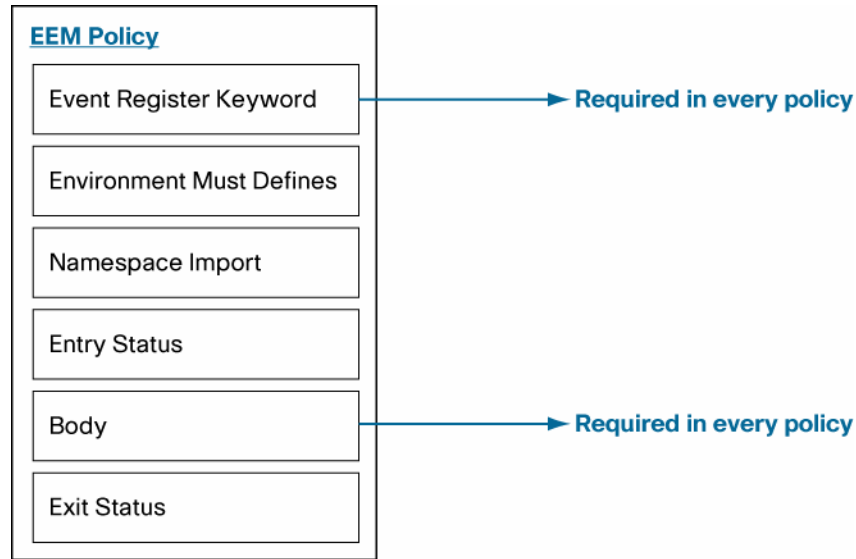
In general, each EEM policy has two parts:

- An event keyword establishes the criteria used to determine when the policy is run.
- Configured actions in the policy implement recovery response to the detected event.

Programming Policies with TCL

EEM policies require two parts: the event register keyword and the body. The remaining parts of the policy are optional: environment must defines, namespace import, entry status, and exit status (Figure 5).

Figure 5. EEM Policy Parts



Event Register Keyword

The start of every policy must describe and register the event to detect by using the event register keyword.

Environment Must Defines

Environment must defines are variables used in TCL script; they are defined in the router configuration using the event manager environment command.

Namespace Import

Namespace import includes all TCL commands closely related to EEM.

Entry Status

The entry status part of an EEM policy is used to determine if a prior policy has been run for the same event. If the prior policy is successful, the current policy may or may not require execution.

Entry status designations can use one of three possible values: 0 (previous policy was successful), Not=0 (previous policy failed), and undefined (no previous policy was executed).

Body

The body is the actual body of the TCL policy script.

Exit Status

When a policy exits, the exit value is used by the Embedded Event Manager to determine whether or not to apply the default action for this event, if any. A value of zero means “do not perform the default action”. A non-zero value means “perform the default action”.

The exit status is passed to subsequent policies that are run for the same event.

A sample EEM script follows:

```

::cisco::eem::event_register_none Event Register Keyword

```

```

# Namespace imports
namespace import :: isco::eem::*
namespace import :: isco::lib::*

# Local procedure for CLI interface
# Pass a list of cli commands and it returns a list of outputs
#
proc CLICmdProc {cmds} {
  if [catch {cli_open} result] {
    error $result $errorInfo
  } else {
    array set cli1 $result
  }
  if [catch {cli_exec $cli1(fd) "enable"} result] {
    error $result $errorInfo
  }
  if [catch {cli_exec $cli1(fd) "term len 0"} result] {
    error $result $errorInfo
  }
  foreach a_cmd $cmds {
    if [catch {cli_exec $cli1(fd) $a_cmd} result] {
      error $result $errorInfo
    } else {
      lappend cmd_output $result
    }
  }
  if [catch {cli_close $cli1(fd) $cli1(tty_id)} result] {
    error $result $errorInfo
  }
  return $cmd_output
}
#
# Put commands here
#
# enhancement - read them from a file
#
lappend clicmd "show ip int brief"
lappend clicmd "show tech"
lappend clicmd "show clock"
#
set cliout [CLICmdProc $clicmd]
#
# write to a file
#
#
if [file exists $_filename] {
  puts "file $_filename being overwritten"
}
set myfileid [open $_filename w+]

```

Namespace Import

Body

```

foreach outs $cliout {
puts $myfileid $outs
}
close $myfileid

```

EEM Script Support

Cisco IOS Software has built-in EEM sample policies registered that you can use for further script customization according to your requirements. For script support, please contact your accounts team or the reseller.

For more information about new features and functions on EEM, send an e-mail message to rwill@cisco.com.

Additional References

- [EEM Policy Script Community](#)
- [Writing EEM Policies Using TCL](#)



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices

CCDE, CCVP, Cisco EEM, Cisco StadiumView, the Cisco logo, COE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play and Learn is a service mark; and Access Registrar, Aronix, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCR, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Enterprise/Solver Ether/Chemel, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Presence, FrameShare, GigaDrive, HomeLink, Internet Companion, IOS, IPPhone, IPTV, IQ Expertise, the IQ logo, IQ Net, iQ Ready, iQ Ready: See More, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MGX, NetWorker, Networking Academy, Network Registrar, PCNow, PIX, PowerPanel, ProCommand, ScriptGuard, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Power to Increase Your Returns, DuoGuard, TelePresence, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word "partner" does not imply a partnership relationship between Cisco and any other company. (08010)