Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 6.x

First Published: 2013-11-20

Last Modified: 2020-07-09

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

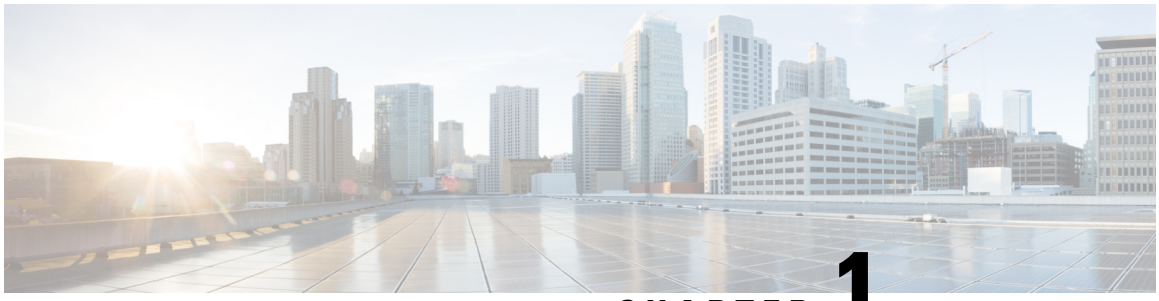
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 6.x*.

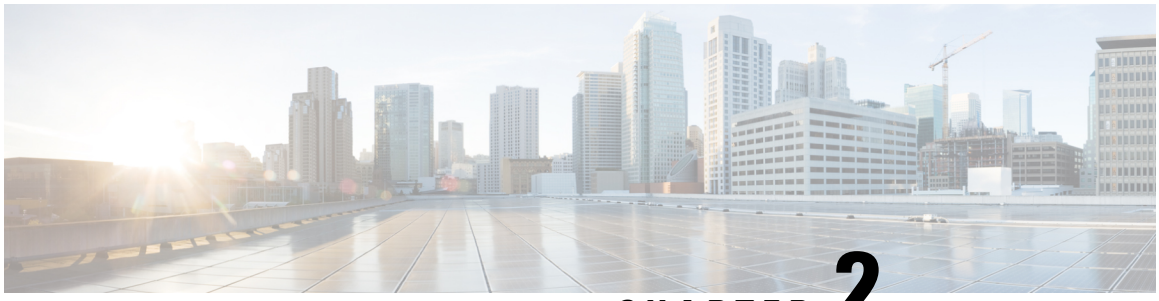
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 6.x* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 6.x

Feature	Description	Changed in Release	Where Documented
Software maintenance upgrades (SMUs)	Added the procedure to perform an SMU for Guest Shell Bash.	Applies to any Guest Shell Bash SMU for 6.1(2)I3(1)	Performing Software Maintenance Upgrades, on page 207
Software maintenance upgrades (SMUs)	Added the requirement to add and activate the SMU package in separate commands.	Applies to the Cisco NX-OS CSCur02700 SMU for 6.1(2)I3(1) and all 6.1(2)I2(x) releases	Performing Software Maintenance Upgrades, on page 207
ERSPAN	Added support for FEX ports as an ERSPAN source.	6.1(2)I2(3)	Configuring ERSPAN, on page 189
SPAN	Added support for FEX ports as a SPAN source.	6.1(2)I2(3)	Configuring SPAN, on page 177

Feature	Description	Changed in Release	Where Documented
Software maintenance upgrades (SMUs)	Added the requirement to reload the standby supervisor module when committing or deactivating the SMU package.	Applies to any Cisco NX-OS SMU for releases prior to 6.1(2)I2(2b)	Performing Software Maintenance Upgrades, on page 207
SNMP	Added support for traffic storm control.	6.1(2)I2(2a)	Configuring SNMP, on page 101
SPAN	Added support for uplink ports as SPAN destinations on Cisco Nexus 9300 Series switches.	6.1(2)I2(2)	Configuring SPAN, on page 177
CDP	Added support for native VLANs, VTP, access ports, and trunk ports.	6.1(2)I2(1)	Configuring CDP, on page 23
DCNM	Introduced this feature.	6.1(2)I2(1)	Overview, on page 3
EEM	Added support for traffic storm control and object tracking.	6.1(2)I2(1)	Configuring the Embedded Event Manager, on page 149
ERSPAN	Added support for source VLANs.	6.1(2)I2(1)	Configuring ERSPAN, on page 189
LLDP	Added support for port VLANs.	6.1(2)I2(1)	Configuring LLDP, on page 201
SNMP	Added support for HSRP, STP, and VTP.	6.1(2)I2(1)	Configuring SNMP, on page 101
Software maintenance upgrades (SMUs)	Introduced this feature.	6.1(2)I2(1)	Performing Software Maintenance Upgrades, on page 207
SPAN	Added support for source VLANs and support for SPAN destination ports in access or trunk mode.	6.1(2)I2(1)	Configuring SPAN, on page 177



CHAPTER 2

Overview

This chapter describes the system management features that you can use to monitor and manage Cisco NX-OS devices.

This chapter contains the following sections:

Software Image

The Cisco NX-OS software consists of one NXOS software image. This image runs on all Cisco Nexus 3400 Series switches.

Licensing Requirements

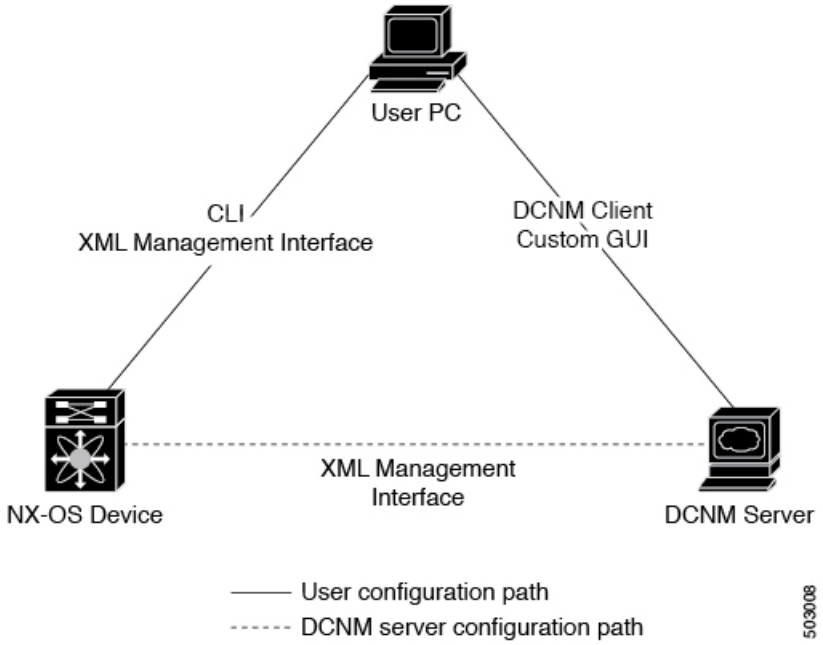
For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Cisco NX-OS Device Configuration Methods

You can configure devices using direct network configuration methods or web services hosted on a Cisco Data Center Network Management (DCNM) server.

This figure shows the device configuration methods available to a network user.

Figure 1: Cisco NX-OS Device Configuration Methods



This table lists the configuration method and the document where you can find more information.

Table 2: Configuration Methods Book Links

Configuration Method	Document
CLI from a Secure Shell (SSH) session, a Telnet session, or the console port	Cisco Nexus 3400 Series NX-OS Fundamentals Configuration Guide
Cisco DCNM client	*Cisco DCNM Fundamentals Guide*

Configuring with CLI or XML Management Interface

You can configure Cisco NX-OS devices using the command-line interface (CLI) or the XML management interface over Secure Shell (SSH) as follows:

- CLI from an SSH session, a Telnet session, or the console port—You can configure devices using the CLI from an SSH session, a Telnet session, or the console port. SSH provides a secure connection to the device. For more information, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide.
- XML management interface over SSH—You can configure devices using the XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI functionality. For more information, see the *Cisco NX-OS XML Management Interface User Guide*.

Configuring with Cisco DCNM

You can configure Cisco NX-OS devices using the Cisco DCNM client, which runs on your local PC and uses web services on the Cisco DCNM server. The Cisco DCNM server configures the device over the XML management interface. For more information about the Cisco DCNM client, see the Cisco DCNM Fundamentals Guide.

Network Time Protocol

The Network Time Protocol (NTP) synchronizes the time of day among a set of distributed time servers and clients so that you can correlate time-specific information, such as system logs, received from the devices in your network.

Cisco Discovery Protocol

You can use the Cisco Discovery Protocol (CDP) to discover and view information about all Cisco equipment that is directly attached to your device. CDP runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. CDP is media and protocol independent, and gathers the protocol addresses of neighboring devices, discovering the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

System Messages

You can use system message logging to control the destination and to filter the severity level of messages that system processes generate. You can configure logging to a terminal session, a log file, and syslog servers on remote systems.

For more information about the system message format and the messages that the device generates, see the *Cisco NX-OS System Messages Reference*.

Smart Call Home

Call Home provides an e-mail-based notification of critical system policies. Cisco NX-OS provides a range of message formats for optimal compatibility with pager services, standard e-mail, or XML-based automated parsing applications. You can use this feature to page a network support engineer, e-mail a Network Operations Center, or use Cisco Smart Call Home services to automatically generate a case with the Technical Assistance Center.

Rollback

The rollback feature allows you to take a snapshot, or checkpoint, of the device configuration and then reapply that configuration at any point without having to reload. Rollback allows any authorized administrator to apply this checkpoint configuration without requiring expert knowledge of the features configured in the checkpoint.

Session Manager allows you to create a configuration session and apply all commands within that session atomically.

Session Manager

Session Manager allows you to create a configuration and apply it in batch mode after the configuration is reviewed and verified for accuracy and completeness.

Scheduler

The scheduler allows you to create and manage jobs such as routinely backing up data or making quality of service (QoS) policy changes. The scheduler can start a job according to your needs—only once at a specified time or at periodic intervals.

SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

RMON

Remote monitoring (RMON) is an Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. Cisco NX-OS supports RMON alarms, events, and logs to monitor Cisco NX-OS devices.

Online Diagnostics

Cisco Generic Online Diagnostics (GOLD) define a common framework for diagnostic operations across Cisco platforms. The online diagnostic framework specifies the platform-independent fault-detection architecture for centralized and distributed systems, including the common diagnostics CLI and the platform-independent fault-detection procedures for boot-up and run-time diagnostics. The platform-specific diagnostics provide hardware-specific fault-detection tests and allow you to take appropriate corrective action in response to diagnostic test results.

Embedded Event Manager

The Embedded Event Manager (EEM) allows you to detect and handle critical events in the system. EEM provides event detection and recovery, including monitoring of events either as they occur or as thresholds are crossed.

Onboard Failure Logging

You can configure a device to log failure data to persistent storage, which you can retrieve and display for analysis at a later time. This on-board failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. This information is useful for analysis of failed modules.

SPAN

You can configure an Ethernet Switched Port Analyzer (SPAN) to monitor traffic in and out of your device. The SPAN features allow you to duplicate packets from source ports to destination ports.

ERSPAN

Encapsulated Remote Switched Port Analyzer (ERSPAN) is used to transport mirrored traffic in an IP network. ERSPAN supports source ports, source VLANs, and destinations on different switches, which provide remote monitoring of multiple switches across your network.

To configure an ERSPAN source session, you associate a set of source ports or VLANs with a destination IP address, ERSPAN ID number, and virtual routing and forwarding (VRF) name.

LLDP

Link Layer Discovery Protocol (LLDP) is a vendor-neutral, one-way device discovery protocol that allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other. You can enable LLDP globally or per interface.

SMUs

A software maintenance upgrade (SMU) is a package file that contains fixes for a specific defect. SMUs are created to respond to immediate issues and do not include new features. SMUs are not an alternative to maintenance releases. They provide a quick resolution of immediate issues. All defects fixed by SMUs are integrated into the maintenance releases.

Virtual Device Contexts

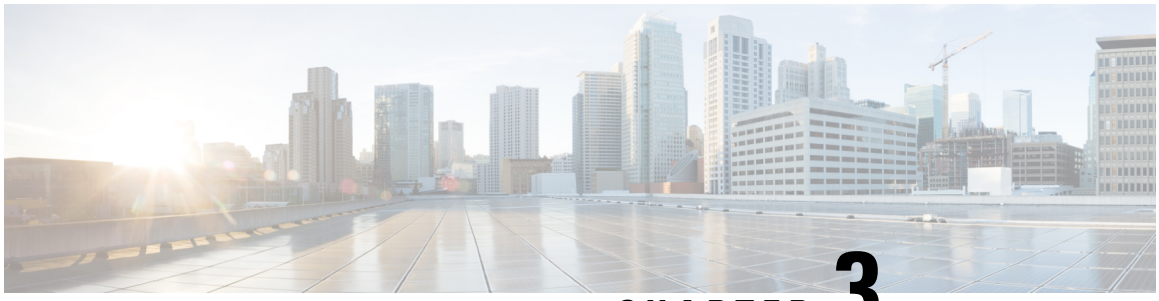
Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 9000 Series switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.

Troubleshooting Features

Cisco NX-OS provides troubleshooting tools such as ping, traceroute, Ethanalyzer, and the Blue Beacon feature.

When a service fails, the system generates information that can be used to determine the cause of the failure. The following sources of information are available:

- Every service restart generates a syslog message of level LOG_ERR.
- If the Smart Call Home service is enabled, every service restart generates a Smart Call Home event.
- If SNMP traps are enabled, the SNMP agent sends a trap when a service is restarted.
- When a service failure occurs on a local module, you can view a log of the event by entering the **show processes log** command in that module. The process logs are persistent across supervisor switchovers and resets.
- When a service fails, a system core image file is generated. You can view recent core images by entering the **show cores** command on the active supervisor. Core files are not persistent across supervisor switchovers and resets, but you can configure the system to export core files to an external server using the file transfer utility Trivial File Transfer Protocol (TFTP) by entering the **system cores** command.
- CISCO-SYSTEM-MIB contains a table for cores (cseSwCoresTable).



CHAPTER 3

Configuring NTP

This chapter describes how to configure the Network Time Protocol (NTP) on Cisco NX-OS devices.

This chapter includes the following sections:

About NTP

The Network Time Protocol (NTP) synchronizes the time of day among a set of distributed time servers and clients so that you can correlate events when you receive system logs and other time-specific events from multiple network devices. NTP uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communications use Coordinated Universal Time (UTC).

An NTP server usually receives its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server, and then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to within a millisecond of each other.

NTP uses a stratum to describe the distance between a network device and an authoritative time source:

- A stratum 1 time server is directly attached to an authoritative time source (such as a radio or atomic clock or a GPS time source).
- A stratum 2 NTP server receives its time through NTP from a stratum 1 time server.

Before synchronizing, NTP compares the time reported by several network devices and does not synchronize with one that is significantly different, even if it is a stratum 1. Because Cisco NX-OS cannot connect to a radio or atomic clock and act as a stratum 1 server, we recommend that you use the public NTP servers available on the Internet. If the network is isolated from the Internet, Cisco NX-OS allows you to configure the time as though it were synchronized through NTP, even though it was not.

Note You can create NTP peer relationships to designate the time-serving hosts that you want your network device to consider synchronizing with and to keep accurate time if a server failure occurs.

The time kept on a device is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

NTP Associations

An NTP association can be one of the following:

- A peer association—The device can either synchronize to another device or allow another device to synchronize to it.
- A server association—The device synchronizes to a server.

You need to configure only one end of an association. The other device can automatically establish the association.

NTP as a Time Server

The Cisco NX-OS device can use NTP to distribute time. Other devices can configure it as a time server. You can also configure the device to act as an authoritative NTP server, enabling it to distribute time even when it is not synchronized to an outside time source.

Clock Manager

Clocks are resources that need to be shared across different processes. Multiple time synchronization protocols, such as NTP, might be running in the system.

The clock manager allows you to specify the protocol to control the various clocks in the system. Once you specify the protocol, the system clock starts updating. For information on configuring the clock manager, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide.

High Availability

Stateless restarts are supported for NTP. After a reboot or a supervisor switchover, the running configuration is applied. For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.

You can configure NTP peers to provide redundancy in case an NTP server fails.

Virtualization Support

NTP recognizes virtual routing and forwarding (VRF) instances. NTP uses the default VRF if you do not configure a specific VRF for the NTP server and NTP peer. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about VRFs.

Prerequisites for NTP

NTP has the following prerequisites:

- To configure NTP, you must have connectivity to at least one server that is running NTP.

Guidelines and Limitations for NTP

NTP has the following configuration guidelines and limitations:

- NTP server functionality is supported.
- You should have a peer association with another device only when you are sure that your clock is reliable (which means that you are a client of a reliable NTP server).
- A peer configured alone takes on the role of a server and should be used as a backup. If you have two servers, you can configure several devices to point to one server and the remaining devices to point to the other server. You can then configure a peer association between these two servers to create a more reliable NTP configuration.
- If you have only one server, you should configure all the devices as clients to that server.
- You can configure up to 64 NTP entities (servers and peers).
- If you configure NTP in a VRF, ensure that the NTP server and peers can reach each other through the configured VRFs.
- You must manually distribute NTP authentication keys on the NTP server and Cisco NX-OS devices across the network.
- If you are using the switch as an edge device and want to use NTP, Cisco recommends using the **ntp access-group** command and filtering NTP only to the required edge devices.
- If the system has been configured with the **ntp passive**, **ntp broadcast client**, or **ntp multicast client** commands, when NTP receives an incoming symmetric active, broadcast, or multicast packet, it can set up an ephemeral peer association in order to synchronize with the sender.
- If the **ntp authenticate** command is specified, when a symmetric active, broadcast, or multicast packet is received, the system does not synchronize to the peer unless the packet carries one of the authentication keys specified in the **ntp trusted-key** global configuration command.
- To prevent synchronization with unauthorized network hosts, the **ntp authenticate** command should be specified any time the **ntp passive**, **ntp broadcast client**, or **ntp multicast client** command has been specified unless other measures, such as the **ntp access-group** command, have been taken to prevent unauthorized hosts from communicating with the NTP service on the device.
- The **ntp authenticate** command does not authenticate peer associations configured via the **ntp server** and **ntp peer** configuration commands. To authenticate the **ntp server** and **ntp peer** associations, specify the **key** keyword.
-

Default Settings for NTP

The following table lists the default settings for NTP parameters.

Parameters	Default
NTP	Enabled
NTP authentication	Disabled
NTP access	Enabled
NTP logging	Disabled

Configuring NTP

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Enabling or Disabling NTP

You can enable or disable NTP. NTP is enabled by default.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature ntp** **Example:** `switch(config)# feature ntp`	Enables or disables NTP.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Device as an Authoritative NTP Server

You can configure the device to act as an authoritative NTP server, enabling it to distribute time even when it is not synchronized to an existing time server.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ntp master** [*stratum*] **Example:** `switch(config)# ntp master`	Configures the device as an authoritative NTP server. You can specify a different stratum level from which NTP clients get their time synchronized. The range is from 1 to 15.
Step 3	(Optional) **show running-config ntp** **Example:** `switch(config)# show running-config ntp`	Displays the NTP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an NTP Server and Peer

You can configure an NTP server and peer.

Before you begin

Make sure you know the IP address or Domain Name System (DNS) names of your NTP server and its peers.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ntp server** {*ip-address* | *ipv6-address* | *dns-name*} [**key** *key-id*] [**maxpoll** *max-poll*] [**minpoll** *min-poll*] [**prefer**] [**use-vrf** *vrf-name*] **Example:** `switch(config)# ntp server 192.0.2.10`	Forms an association with a server. Use the **key** keyword to configure a key to be used while communicating with the NTP server. The range for the *key-id* argument is from 1 to 65535. Use the **maxpoll** and **minpoll** keywords to configure the maximum and minimum intervals in which to poll a server. The range for the

	Command or Action	Purpose
		max-poll and *min-poll* arguments is from 4 to 16 (configured as powers of 2, so effectively 16 to 65536 seconds), and the default values are 6 and 4, respectively (*maxpoll* default = 64 seconds, *minpoll* default = 16 seconds). Use the **prefer** keyword to make this server the preferred NTP server for the device. Use the **use-vrf** keyword to configure the NTP server to communicate over the specified VRF. The *vrf-name* argument can be **default** , **management** , or any case-sensitive, alphanumeric string up to 32 characters. **Note** If you configure a key to be used while communicating with the NTP server, make sure that the key exists as a trusted key on the device.
Step 3	[**no**] **ntp peer** {*ip-address* | *ipv6-address* | *dns-name*} [**key** *key-id*] [**maxpoll** *max-poll*] [**minpoll** *min-poll*] [**prefer**] [**use-vrf** *vrf-name*] **Example:** `switch(config)# ntp peer 2001:0db8::4101`	Forms an association with a peer. You can specify multiple peer associations. Use the **key** keyword to configure a key to be used while communicating with the NTP peer. The range for the *key-id* argument is from 1 to 65535. Use the **maxpoll** and **minpoll** keywords to configure the maximum and minimum intervals in which to poll a server. The range for the *max-poll* and *min-poll* arguments is from 4 to 17 (configured as powers of 2, so effectively 16 to 131072 seconds), and the default values are 6 and 4, respectively (*maxpoll* default = 64 seconds, *minpoll* default = 16 seconds). Use the **prefer** keyword to make this peer the preferred NTP peer for the device. Use the **use-vrf** keyword to configure the NTP peer to communicate over the specified VRF. The *vrf-name* argument can be **default** , **management** , or any case-sensitive, alphanumeric string up to 32 characters.
Step 4	(Optional) **show ntp peers** **Example:**	Displays the configured server and peers.

	Command or Action	Purpose
	`switch(config)# show ntp peers`	**Note** A domain name is resolved only when you have a DNS server configured. When DNS/Name Server resolves both IPv4 and IPv6, IPv6 Address is preferred by NX-OS.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring NTP Authentication

You can configure the device to authenticate the time sources to which the local clock is synchronized. When you enable NTP authentication, the device synchronizes to a time source only if the source carries one of the authentication keys specified by the **ntp trusted-key** command. The device drops any packets that fail the authentication check and prevents them from updating the local clock. NTP authentication is disabled by default.

Before you begin

Make sure that you configured the NTP server with the authentication keys that you plan to specify in this procedure.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ntp authentication-key** *number* **md5** *md5-string* **Example:** `switch(config)# ntp authentication-key 42 md5 aNiceKey`	Defines the authentication keys. The device does not synchronize to a time source unless the source has one of these authentication keys and the key number is specified by the **ntp trusted-key** *number* command. The range for authentication keys is from 1 to 65535. For the MD5 string, you can enter up to eight alphanumeric characters.
Step 3	**ntp server** *ip-address* **key** *key-id* **Example:**	Forms an association with a server. Use the **key** keyword to configure a key to be used while communicating with the NTP server.

	Command or Action	Purpose
	`switch(config)# ntp server 192.0.2.1 key 1001`	The range for the *key-id* argument is from 1 to 65535. To require authentication, the **key** keyword must be used. Any **ntp server** or **ntp peer** commands that do not specify the **key** keyword will continue to operate without authentication.
Step 4	(Optional) **show ntp authentication-keys** **Example:** `switch(config)# show ntp authentication-keys`	Displays the configured NTP authentication keys.
Step 5	[**no**] **ntp trusted-key** *number* **Example:** `switch(config)# ntp trusted-key 42`	Specifies one or more keys (defined in Step 2) that an unconfigured remote symmetric, broadcast, and multicast time source must provide in its NTP packets in order for the device to synchronize to it. The range for trusted keys is from 1 to 65535. This command provides protection against accidentally synchronizing the device to a time source that is not trusted.
Step 6	(Optional) **show ntp trusted-keys** **Example:** `switch(config)# show ntp trusted-keys`	Displays the configured NTP trusted keys.
Step 7	[**no**] **ntp authenticate** **Example:** `switch(config)# ntp authenticate`	Enables or disables authentication for ntp passive, ntp broadcast client, and ntp multicast. NTP authentication is disabled by default.
Step 8	(Optional) **show ntp authentication-status** **Example:** `switch(config)# show ntp authentication-status`	Displays the status of NTP authentication.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring NTP Access Restrictions

You can control access to NTP services by using access groups. Specifically, you can specify the types of requests that the device allows and the servers from which it accepts responses.

If you do not configure any access groups, NTP access is granted to all devices. If you configure any access groups, NTP access is granted only to the remote device whose source IP address passes the access list criteria.

- Without the **match-all** keyword, the packet gets evaluated against the access groups (in the order mentioned below) until it finds a permit. If a permit is not found, the packet is dropped.
- With **match-all** keyword, the packet gets evaluated against all the access groups (in the order mentioned below) and the action is taken based on the last successful evaluation (the last access group where an ACL is configured).

- peer—process client, symmetric active, symmetric passive, serve, control, and private packets(all types)
- serve—process client, control, and private packets
- serve-only—process client packets only
- query-only—process control and private packets only

The access groups are evaluated in the following order:

1. peer (all packet types)
2. serve (client, control, and private packets)
3. serve-only (client packets) or query-only (control and private packets)

ACL processing of serve-only or query-only depends on the NTP packet type.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show ntp access-groups** **Example:** `switch(config)# show ntp access-groups`	Displays the NTP access group configuration.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the NTP Source IP Address

NTP sets the source IP address for all NTP packets based on the address of the interface through which the NTP packets are sent. You can configure NTP to use a specific source IP address.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ntp source** *ip-address* **Example:** `switch(config)# ntp source 192.0.2.1`	Configures the source IP address for all NTP packets. The *ip-address* can be in IPv4 or IPv6 format.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the NTP Source Interface

You can configure NTP to use a specific interface.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ntp source-interface** *interface* **Example:** `switch(config)# ntp source-interface ethernet 2/1`	Configures the source interface for all NTP packets. Use the **?** keyword to display a list of supported interfaces.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring NTP Logging

You can configure NTP logging in order to generate system logs with significant NTP events. NTP logging is disabled by default.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ntp logging** **Example:** `switch(config)# ntp logging`	Enables or disables system logs to be generated with significant NTP events. NTP logging is disabled by default.
Step 3	(Optional) **show ntp logging-status** **Example:** `switch(config)# show ntp logging-status`	Displays the NTP logging configuration status.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the NTP Configuration

To display the NTP configuration, perform one of the following tasks:

Command	Purpose
show ntp access-groups	Displays the NTP access group configuration.
show ntp authentication-keys	Displays the configured NTP authentication keys.
show ntp authentication-status	Displays the status of NTP authentication.
show ntp logging-status	Displays the NTP logging status.
show ntp peer-status	Displays the status for all NTP servers and peers.
show ntp peers	Displays all the NTP peers.
show ntp rts-update	Displays the RTS update status.
show ntp source	Displays the configured NTP source IP address.
show ntp source-interface	Displays the configured NTP source interface.
show ntp statistics {**io** | **local** | **memory** | **peer** {**ipaddr** {*ipv4-addr* | *ipv6-addr*} | **name** *peer-name*}}	Displays the NTP statistics.
show ntp trusted-keys	Displays the configured NTP trusted keys.
show running-config ntp	Displays NTP information.

Use the **clear ntp session** command to clear the NTP sessions.

Use the **clear ntp statistics** command to clear the NTP statistics.

Configuration Examples for NTP

This example shows how to configure the device to synchronize only to time sources that provide authentication key 42 in their NTP packets:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ntp authentication-key 42 md5 aNiceKey
switch(config)# ntp server 192.0.2.105 key 42
switch(config)# ntp trusted-key 42
switch(config)# ntp authenticate
switch(config)# copy running-config startup-config
[########################################] 100%
switch(config)#
```

This example shows an NTP access group configuration with the following restrictions:

- Peer restrictions are applied to IP addresses that pass the criteria of the access list named “peer-acl.”
- Serve restrictions are applied to IP addresses that pass the criteria of the access list named “serve-acl.”
- Serve-only restrictions are applied to IP addresses that pass the criteria of the access list named “serve-only-acl.”
- Query-only restrictions are applied to IP addresses that pass the criteria of the access list named “query-only-acl.”

```
switch# configure terminal
switch(config)# ntp peer 10.1.1.1
switch(config)# ntp peer 10.2.2.2
switch(config)# ntp peer 10.3.3.3
switch(config)# ntp peer 10.4.4.4
switch(config)# ntp peer 10.5.5.5
switch(config)# ntp peer 10.6.6.6
switch(config)# ntp peer 10.7.7.7
switch(config)# ntp peer 10.8.8.8
switch(config)# ntp access-group peer peer-acl
switch(config)# ntp access-group serve serve-acl
switch(config)# ntp access-group serve-only serve-only-acl
switch(config)# ntp access-group query-only query-only-acl
switch(config)# ip access-list peer-acl
switch(config-acl)# 10 permit ip host 10.1.1.1 any
switch(config-acl)# 20 permit ip host 10.8.8.8 any
switch(config)# ip access-list serve-acl
switch(config-acl)# 10 permit ip host 10.4.4.4 any
switch(config-acl)# 20 permit ip host 10.5.5.5 any
switch(config)# ip access-list serve-only-acl
switch(config-acl)# 10 permit ip host 10.6.6.6 any
switch(config-acl)# 20 permit ip host 10.7.7.7 any
switch(config)# ip access-list query-only-acl
switch(config-acl)# 10 permit ip host 10.2.2.2 any
switch(config-acl)# 20 permit ip host 10.3.3.3 any
```

Note When only a single ACL group is applied, then all the packets relevant for other ACL categories are denied and only packets relevant for the configured ACL group is processed, as mentioned in below scenarios:

- If serve ACL is configured, then only client, control, and private packets are processed and all the other packets are denied.
- If serve-only ACL is configured, then only client packets are processed and all the other packets are denied.

If more than a single ACL is configured, it follows the order of processing as mentioned in below scenario:

- If serve and serve-only both are configured for the same IP address without match-all configured, where the IP is permitted in serve-acl and denied in serve-only, the client, control, private packets are permitted for that IP.

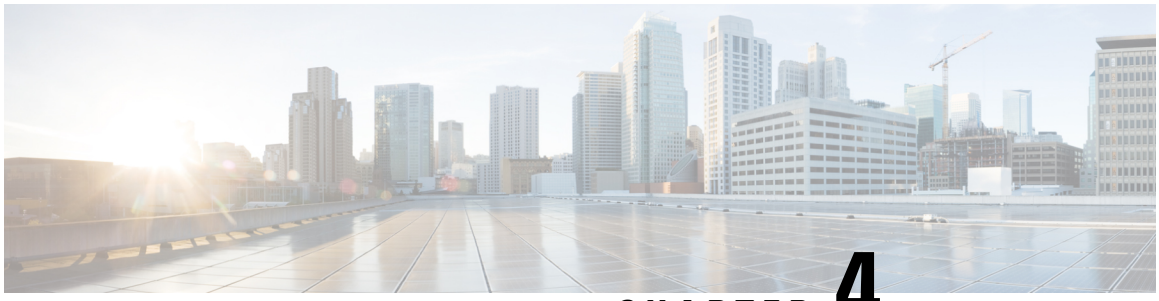
Additional References

Related Documents

Related Topic	Document Title
Clock manager	Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide

MIBs

MIBs	MIBs Link
MIBs related to NTP	To locate and download supported MIBs, go to the follow ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 4

Configuring CDP

This chapter describes how to configure the Cisco Discovery Protocol (CDP) on Cisco NX-OS devices.

This chapter includes the following sections:

About CDP

The Cisco Discovery Protocol (CDP) is a media-independent and protocol-independent protocol that runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. You can use CDP to discover and view information about all the Cisco devices that are directly attached to the device.

CDP gathers protocol addresses of neighboring devices and discovers the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

Each device that you configure for CDP sends periodic advertisements to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain hold-time information, which indicates the length of time that a receiving device should hold CDP information before removing it. You can configure the advertisement or refresh timer and the hold timer.

CDP Version-2 (CDPv2) allows you to track instances where the native VLAN ID or port duplex states do not match between connecting devices.

CDP advertises the following type-length-value fields (TLVs):

- Device ID
- Address
- Port ID
- Capabilities
- Version

- Platform
- Native VLAN
- Full or Half Duplex
- MTU
- SysName
- SysObjectID
- Management Address
- Physical Location
- VTP

All CDP packets include a VLAN ID. If you configure CDP on a Layer 2 access port, the CDP packets sent from that access port include the access port VLAN ID. If you configure CDP on a Layer 2 trunk port, the CDP packets sent from that trunk port include the lowest configured VLAN ID allowed on that trunk port. The trunk port can receive CDP packets that include any VLAN ID in the allowed VLAN list for that trunk port. For more information on VLANs, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

VTP Feature Support

CDP sends the VLAN Trunking Protocol (VTP) type-length-value field (TLV) if the following conditions are met:

- CDP Version 2 is enabled.
- The VTP feature is enabled.
- A VTP domain name is configured.

You can view the VTP information with the **show cdp neighbors detail** command.

High Availability

Cisco NX-OS supports both stateful and stateless restarts and switchover for CDP. For more information on high availability, see the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*.

Virtualization Support

Cisco NX-OS supports one instance of CDP.

Guidelines and Limitations for CDP

CDP has the following configuration guidelines and limitations:

- CDP can discover up to 256 neighbors per port if the port is connected to a hub with 256 connections.

- CDP must be enabled on the device or you cannot enable it on any interfaces.
- You can configure CDP on physical interfaces and port channels only.

Default Settings for CDP

This table lists the default settings for CDP parameters.

Parameters	Default
CDP	Enabled globally and on all interfaces
CDP version	Version 2
CDP device ID	Serial number
CDP timer	60 seconds
CDP hold timer	180 seconds

Configuring CDP

Note The Cisco NX-OS commands for this feature may differ from those commands that are used in Cisco IOS.

Enabling or Disabling CDP Globally

CDP is enabled by default. You can disable CDP and then reenable it.

You must enable CDP on the device before you enable CDP on any interfaces. If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **cdp enable** **Example:** `switch(config)# cdp enable`	Enables or disables the CDP feature on the entire device. It is enabled by default.
Step 3	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Enabling or Disabling CDP on an Interface

CDP is enabled by default on an interface. You can disable CDP on an interface.

If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot/port*
	Example:
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**] **cdp enable** **Example:** `switch(config-if)# cdp enable`	Enables or disables CDP on this interface. It is enabled by default. **Note** Make sure that CDP is enabled globally on the device.
Step 4	(Optional) **show cdp interface** *interface slot/port* **Example:** `switch(config-if)# show cdp interface ethernet 1/2`	Displays CDP information for an interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Optional CDP Parameters

You can use the optional commands in this procedure to modify CDP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **cdp advertise** {**v1** | **v2**} **Example:** `switch(config)# cdp advertise v1`	Sets the CDP version that is supported by the device. The default is v2.
Step 3	(Optional) **cdp format device-id** {**mac-address** | **serial-number** | **system-name**} **Example:** `switch(config)# cdp format device-id mac-address`	Sets the CDP device ID. The options are as follows: • **mac-address**—The MAC address of the chassis. • **serial-number**—The chassis serial number/Organizationally Unique Identifier (OUI). • **system-name**—The system name or fully qualified domain name. The default is **system-name**.
Step 4	(Optional) **cdp holdtime** *seconds* **Example:** `switch(config)# cdp holdtime 150`	Sets the time that CDP holds onto neighbor information before removing it. The range is from 10 to 255 seconds. The default is 180 seconds.
Step 5	(Optional) **cdp timer** *seconds* **Example:** `switch(config)# cdp timer 50`	Sets the refresh time when CDP sends advertisements to neighbors. The range is from 5 to 254 seconds. The default is 60 seconds.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the CDP Configuration

To display the CDP configuration, perform one of the following tasks:

Command	Purpose
show cdp all	Displays all interfaces that have CDP enabled.
show cdp entry {**all** | **name** *entry-name*}	Displays the CDP database entries.

Command	Purpose
show cdp global	Displays the CDP global parameters.
show cdp interface *interface slot/port*	Displays the CDP interface status.
show cdp neighbors {**device-id** | **interface** *interface slot/port*} [**detail**]	Displays the CDP neighbor status.
show cdp interface *interface slot/port*	Displays the CDP traffic statistics on an interface.

Use the **clear cdp counters** command to clear CDP statistics on an interface.

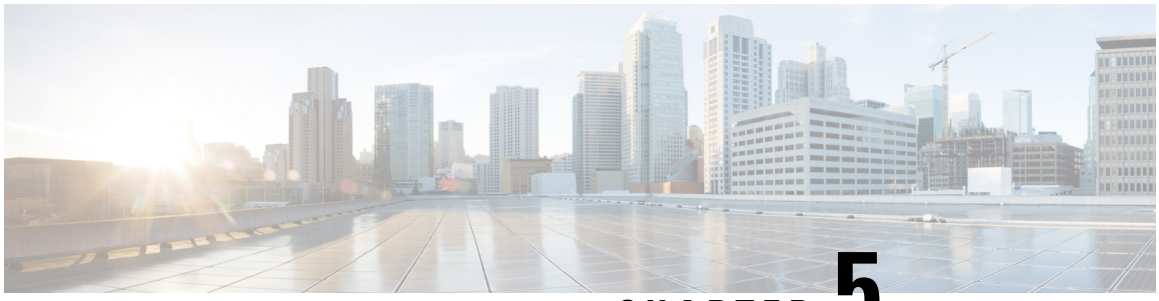
Use the **clear cdp table** command to clear the CDP cache for one or all interfaces.

It is recommended to use the **show cdp neighbors detail** command instead of **show cdp neighbors** command. The **show cdp neighbors** command can display only 13 characters of a platform name. To get the full platform name in the display, use **show cdp neighbors detail** command.

Configuration Example for CDP

This example shows how to enable the CDP feature and configure the refresh and hold timers:

```
configure terminal
cdp enable
cdp timer 50
cdp holdtime 100
```



CHAPTER 5

Configuring System Message Logging

This chapter describes how to configure system message logging on Cisco NX-OS devices.

This chapter contains the following sections:

About System Message Logging

You can use system message logging to control the destination and to filter the severity level of messages that system processes generate. You can configure logging to terminal sessions, a log file, and syslog servers on remote systems.

For more information about the system message format and the messages that the device generates, see the Cisco NX-OS System Messages Reference.

By default, the device outputs messages to terminal sessions and logs system messages to a log file.

The following table describes the severity levels used in system messages. When you configure the severity level, the system outputs messages at that level and lower.

Table 3: System Message Severity Levels

Level	Description
0 – emergency	System unusable
1 – alert	Immediate action needed
2 – critical	Critical condition
3 – error	Error condition
4 – warning	Warning condition

Level	Description
5 – notification	Normal but significant condition
6 – informational	Informational message only
7 – debugging	Appears during debugging only

The device logs the most recent 100 messages of severity 0, 1, or 2 to the NVRAM log. You cannot configure logging to the NVRAM.

You can configure which system messages should be logged based on the facility that generated the message and its severity level.

Syslog Servers

The syslog servers run on remote systems that log system messages based on the syslog protocol. You can configure up to eight IPv4 or IPv6 syslog servers.

To support the same configuration of syslog servers on all switches in a fabric, you can use Cisco Fabric Services (CFS) to distribute the syslog server configuration.

Note When the device first initializes, messages are sent to syslog servers only after the network is initialized.

Guidelines and Limitations for System Message Logging

System message logging has the following configuration guidelines and limitations:

- System messages are logged to the console and the log file by default.
- Any system messages that are printed before the syslog server is reachable (such as supervisor active or online messages) cannot be sent to the syslog server.
- Generally, the syslogs display the local time zone. However, few components such as NGINX display the logs in UTC time zone.

Default Settings for System Message Logging

The following table lists the default settings for the system message logging parameters.

Table 4: Default System Message Logging Parameters

Parameters	Default
Console logging	Enabled at severity level 2
Monitor logging	Enabled at severity level 5

Parameters	Default
Log file logging	Enabled to log messages at severity level 5
Module logging	Enabled at severity level 5
Facility logging	Enabled
Time-stamp units	Seconds
Syslog server logging	Disabled
Syslog server configuration distribution	Disabled

Configuring System Message Logging

Note Be aware that the Cisco NX-OS commands for this feature might differ from those commands used in Cisco IOS.

Configuring System Message Logging to Terminal Sessions

You can configure the device to log messages by their severity level to console, Telnet, and SSH sessions.

By default, logging is enabled for terminal sessions.

Note The current critical (default) logging level is maintained if the console baud speed is 9600 baud (default). All attempts to change the console logging level will generate an error message. To increase the logging level (above critical), you must change the console baud speed to 38400 baud.

Procedure

	Command or Action	Purpose
Step 1	**terminal monitor** **Example:** `switch# terminal monitor`	Enables the device to log messages to the console.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	[**no**] **logging console** [*severity-level*] **Example:** `switch(config)# logging console 3`	Configures the device to log messages to the console session based on a specified severity level or higher. A lower number indicates a higher severity level. Severity levels range from 0 to 7: • 0 – emergency • 1 – alert • 2 – critical • 3 – error • 4 – warning • 5 – notification • 6 – informational • 7 – debugging If the severity level is not specified, the default of 2 is used. The **no** option disables the device's ability to log messages to the console.
Step 4	(Optional) **show logging console** **Example:** `switch(config)# show logging console`	Displays the console logging configuration.
Step 5	[**no**] **logging monitor** [*severity-level*] **Example:** `switch(config)# logging monitor 3`	Enables the device to log messages to the monitor based on a specified severity level or higher. A lower number indicates a higher severity level. Severity levels range from 0 to 7: • 0 – emergency • 1 – alert • 2 – critical • 3 – error • 4 – warning • 5 – notification • 6 – informational • 7 – debugging The configuration applies to Telnet and SSH sessions.

	Command or Action	Purpose
		If the severity level is not specified, the default of 2 is used. The **no** option disables the device's ability to log messages to the Telnet and SSH sessions.
Step 6	(Optional) **show logging monitor** **Example:** `switch(config)# show logging monitor`	Displays the monitor logging configuration.
Step 7	[**no**] **logging message interface type ethernet description** **Example:** `switch(config)# logging message interface type ethernet description`	Enables you to add the description for physical Ethernet interfaces and subinterfaces in the system message log. The description is the same description that was configured on the interface. The **no** option disables the printing of the interface description in the system message log for physical Ethernet interfaces.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Logging System Messages to a File

You can configure the device to log system messages to a file. By default, system messages are logged to the file /logflash/log/*logfilename*.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **logging logfile** *logfile-name severity-level* [| **size** *bytes*] **Example:** `switch(config)# logging logfile my_log 6`	Configures the nonpersistent log file parameters. *logfile-name*: Configures the name of the log file that is used to store system messages. Default filename is "message". *severity-level*: Configures the minimum severity level to log. A lower number indicates a higher severity level. Default is 5. Range is from 0 through 7: • 0 – emergency • 1 – alert

	Command or Action	Purpose
		• 2 – critical • 3 – error • 4 – warning • 5 – notification • 6 – informational • 7 – debugging **size** *bytes*: Optionally specify maximum file size. Range is from 4096 through 4194304 bytes.
Step 3	**logging event** {**link-status** | **trunk-status**} {**enable** | **default**} **Example:** `switch(config)# logging event link-status default`	Logs interface events. • **link-status**—Logs all UP/DOWN and CHANGE messages. • **trunk-status**—Logs all TRUNK status messages. • **enable**—Specifies to enable logging to override the port level configuration. • **default**—Specifies that the default logging configuration is used by interfaces that are not explicitly configured.
Step 4	(Optional) **show logging info** **Example:** `switch(config)# show logging info`	Displays the logging configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Module and Facility Messages Logging

You can configure the severity level and time-stamp units of messages logged by modules and facilities.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	[**no**] **logging module** [*severity-level*] **Example:** `switch(config)# logging module 3`	Enables module log messages that have the specified severity level or higher. Severity levels range from 0 to 7: • 0 – emergency • 1 – alert • 2 – critical • 3 – error • 4 – warning • 5 – notification • 6 – informational • 7 – debugging If the severity level is not specified, the default of 5 is used. The **no** option disables module log messages.
Step 3	(Optional) **show logging module** **Example:** `switch(config)# show logging module`	Displays the module logging configuration.
Step 4	[**no**] **logging level** *facility severity-level* **Example:** `switch(config)# logging level aaa 2`	Enables logging messages from the specified facility that have the specified severity level or higher. Severity levels range from 0 to 7: • 0 – emergency • 1 – alert • 2 – critical • 3 – error • 4 – warning • 5 – notification • 6 – informational • 7 – debugging To apply the same severity level to all facilities, use the **all** facility. For defaults, see the **show logging level** command. The **no** option resets the logging severity level for the specified facility to its default level. If

	Command or Action	Purpose
		you do not specify a facility and severity level, the device resets all facilities to their default levels.
Step 5	(Optional) **show logging level** [*facility*] **Example:** `switch(config)# show logging level aaa`	Displays the logging level configuration and the system default level by facility. If you do not specify a facility, the device displays levels for all facilities.
Step 6	(Optional) [**no**] **logging level** *ethpm* **Example:** **switch(config)# logging level ethpm ?** <0-7> 0-emerg;1-alert;2-crit;3-err;4-warn;5-notif;6-inform;7-debug link-down Configure logging level for link down syslog messages link-up Configure logging level for link up syslog messages **switch(config)#logging level ethpm link-down ?** error ERRORS notif NOTICE **(config)# logging level ethpm link-down error ?** <CR> **(config)# logging level ethpm link-down notif ?** <CR> **switch(config)#logging level ethpm link-up ?** error ERRORS notif NOTICE **(config)# logging level ethpm link-up error ?** <CR> **(config)# logging level ethpm link-up notif ?** <CR>	Enables logging of the Ethernet Port Manager link-up/link-down syslog messages at level 3. Use the **no** option to use the default logging level for Ethernet Port Manager syslog messages.
Step 7	[**no**] **logging timestamp** {**microseconds** | **milliseconds** | **seconds**} **Example:** `switch(config)# logging timestamp milliseconds`	Sets the logging time-stamp units. By default, the units are seconds. **Note** This command applies to logs that are kept in the switch. It does not apply to the external logging server.

	Command or Action	Purpose
Step 8	(Optional) **show logging timestamp** **Example:** `switch(config)# show logging timestamp`	Displays the logging time-stamp units configured.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Syslog Servers

Note Cisco recommends that you configure the syslog server to use the management virtual routing and forwarding (VRF) instance. For more information on VRFs, see Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

You can configure up to eight syslog servers that reference remote systems where you want to log system messages.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **logging server** *host* [*severity-level* [**use-vrf** *vrf-name*]]
	Example:
	`switch(config)# logging server 192.0.2.253`
	Example:
	`switch(config)# logging server 2001::35 use-vrf red`
		Configures a syslog server at the specified hostname, IPv4, or IPv6 address. You can specify logging of messages to a particular syslog server in a VRF by using the **use-vrf** keyword. Severity levels range from 0 to 7:
		• 0 – emergency
		• 1 – alert
		• 2 – critical
		• 3 – error
		• 4 – warning
		• 5 – notification
		• 6 – informational
		• 7 – debugging

	Command or Action	Purpose
		The default outgoing facility is local7. The **no** option removes the logging server for the specified host. The first example forwards all messages on facility local 7. The second example forwards messages with severity level 5 or lower to the specified IPv6 address in VRF red.
Step 3	Required: **logging source-interface loopback** *virtual-interface* **Example:** `switch(config)# logging source-interface loopback 5`	Enables a source interface for the remote syslog server. The range for the *virtual-interface* argument is from 0 to 1023.
Step 4	(Optional) **show logging server** **Example:** `switch(config)# show logging server`	Displays the syslog server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Syslog Servers on a UNIX or Linux System

You can configure a syslog server on a UNIX or Linux system by adding the following line to the /etc/syslog.conf file:

facility.level <five tab characters> *action*

The following table describes the syslog fields that you can configure.

Table 5: Syslog fields in syslog.conf

Field	Description
Facility	Creator of the message, which can be auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, syslog, user, local0 through local7, or an asterisk (*) for all. These facility designators allow you to control the destination of messages based on their origin. **Note** Check your configuration before using a local facility.

Field	Description
Level	Minimum severity level at which messages are logged, which can be debug, info, notice, warning, err, crit, alert, emerg, or an asterisk (*) for all. You can use none to disable a facility.
Action	Destination for messages, which can be a filename, a hostname preceded by the at sign (@), a comma-separated list of users, or an asterisk (*) for all logged-in users.

Procedure

Step 1 Log debug messages with the local7 facility in the file /var/log/myfile.log by adding the following line to the /etc/syslog.conf file:

Example:

```
debug.local7 var/log/myfile.log
```

Step 2 Create the log file by entering these commands at the shell prompt:

Example:

```
$ touch /var/log/myfile.log
$ chmod 666 /var/log/myfile.log
```

Step 3 Make sure the system message logging daemon reads the new changes by checking myfile.log after entering this command:

Example:

```
$ kill -HUP ~cat /etc/syslog.pid~
```

Displaying and Clearing Log Files

You can display or clear messages in the log file and the NVRAM.

Procedure

	Command or Action	Purpose
Step 1	Required: **show logging last** *number-lines* **Example:** `switch# show logging last 40`	Displays the last number of lines in the logging file. You can specify from 1 to 9999 for the last number of lines.
Step 2	**show logging logfile duration** *hh:mm:ss* **Example:**	Displays the messages in the log file that have occurred within the duration entered.

	Command or Action	Purpose
	`switch# show logging logfile duration 15:10:0`	
Step 3	**show logging logfile last-index** **Example:** `switch# show logging logfile last-index`	Displays the sequence number of the last message in the log file.
Step 4	**show logging logfile** [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*] **Example:** `switch# show logging logfile start-time 2013 oct 1 15:10:0`	Displays the messages in the log file that have a timestamp within the span entered. If you do not enter an end time, the current time is used. You enter three characters for the month time field and digits for the year and day time fields.
Step 5	**show logging logfile** [**start-seqn** *number*] [**end-seqn** *number*] **Example:** `switch# show logging logfile start-seqn 100 end-seqn 400`	Displays messages occurring within a range of sequence numbers. If you do not include an end sequence number, the system displays messages from the start number to the last message in the log file.
Step 6	**show logging nvram** [**last** *number-lines*] **Example:** `switch# show logging nvram last 10`	Displays the messages in the NVRAM. To limit the number of lines displayed, you can enter the last number of lines to display. You can specify from 1 to 100 for the last number of lines.
Step 7	**clear logging logfile** [**persistent**] **Example:** `switch# clear logging logfile`	Clears the contents of the log file. **persistent**: Clears the contents of the log file from the persistent location.
Step 8	**clear logging nvram** **Example:** `switch# clear logging nvram`	Clears the logged messages in NVRAM.

Verifying the System Message Logging Configuration

To display system message logging configuration information, perform one of the following tasks:

Command	Purpose
show logging console	Displays the console logging configuration.
show logging info	Displays the logging configuration.
show logging last *number-lines*	Displays the last number of lines of the log file.
show logging level [*facility*]	Displays the facility logging severity level configuration.
show logging logfile duration *hh:mm:ss*	Displays the messages in the log file that have occurred within the duration entered.

Command	Purpose
show logging logfile last-index	Displays the sequence number of the last message in the log file.
show logging logfile [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*]	Displays the messages in the log file based on a start and end date/time.
show logging logfile [**start-seqn** *number*] [**end-seqn** *number*]	Displays messages occurring within a range of sequence numbers. If you do not include an end sequence number, the system displays messages from the start number to the last message in the log file.
show logging module	Displays the module logging configuration.
show logging monitor	Displays the monitor logging configuration.
show logging nvram [**last** *number-lines*]	Displays the messages in the NVRAM log.
show logging server	Displays the syslog server configuration.
show logging timestamp	Displays the logging time-stamp units configuration.

Configuration Example for System Message Logging

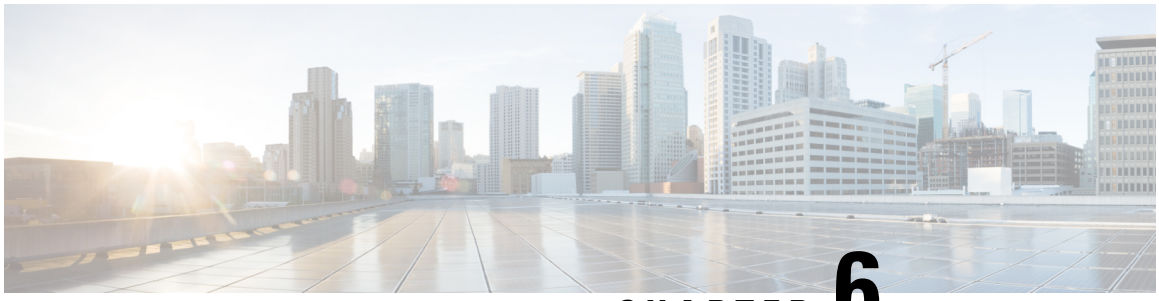
This example shows how to configure system message logging:

```
configure terminal
 logging console 3
 logging monitor 3
 logging logfile my_log 6
 logging module 3
 logging level aaa 2
 logging timestamp milliseconds
 logging server 172.28.254.253
 logging server 172.28.254.254 5 facility local3
 copy running-config startup-config
```

Additional References

Related Documents

Related Topic	Document Title
System messages	*Cisco NX-OS System Messages Reference*



CHAPTER 6

Configuring Smart Call Home

This chapter describes how to configure the Smart Call Home feature of the Cisco NX-OS devices.

This chapter contains the following sections:

About Smart Call Home

Smart Call Home provides an email-based notification for critical system policies. A range of message formats are available for compatibility with pager services, standard email, or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco Smart Call Home services to automatically generate a case with the Technical Assistance Center.

Smart Call Home offers the following features:

- Automatic execution and attachment of relevant CLI command output.
- Multiple message format options such as the following:
  - Short Text—Suitable for pagers or printed reports.
  - Full Text—Fully formatted message information suitable for human reading.
  - XML—Machine-readable format that uses Extensible Markup Language (XML) and Adaptive Messaging Language (AML) XML schema definition (XSD). The AML XSD is published on the Cisco.com website. The XML format enables communication with the Technical Assistance Center.
- Multiple concurrent message destinations. You can configure up to 50 email destination addresses for each destination profile.

Destination Profiles

A destination profile includes the following information:

- One or more alert groups—The group of alerts that trigger a specific Smart Call Home message if the alert occurs.
- One or more email destinations—The list of recipients for the Smart Call Home messages generated by alert groups assigned to this destination profile.
- Message format—The format for the Smart Call Home message (short text, full text, or XML).
- Message severity level—The Smart Call Home severity level that the alert must meet before Cisco NX-OS generates a Smart Call Home message to all email addresses in the destination profile. Cisco NX-OS does not generate an alert if the Smart Call Home severity level of the alert is lower than the message severity level set for the destination profile.

You can also configure a destination profile to allow periodic inventory update messages by using the inventory alert group that will send out periodic messages daily, weekly, or monthly.

Cisco NX-OS supports the following predefined destination profiles:

- CiscoTAC-1—Supports the Cisco-TAC alert group in XML message format. This profile is preconfigured with the callhome@cisco.com email contact, maximum message size, and message severity level 0. You cannot change any of the default information for this profile.
- full-text-destination—Supports the full text message format.
- short-text-destination—Supports the short text message format.

Smart Call Home Alert Groups

An alert group is a predefined subset of Smart Call Home alerts that are supported in all Cisco Nexus devices. Alert groups allow you to select the set of Smart Call Home alerts that you want to send to a predefined or custom destination profile. The device sends Smart Call Home alerts to email destinations in a destination profile only if that Smart Call Home alert belongs to one of the alert groups associated with that destination profile and if the alert has a Smart Call Home message severity at or above the message severity set in the destination profile.

The following table lists the supported alert groups and the default CLI command output included in Smart Call Home messages generated for the alert group.

Table 6: Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Cisco-TAC	All critical alerts from the other alert groups destined for Smart Call Home.	Execute commands based on the alert group that originates the alert.
Configuration	Periodic events related to configuration.	**show module** **show version**

Alert Group	Description	Executed Commands
Diagnostic	Events generated by diagnostics.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
EEM	Events generated by EEM.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show module** **show tech-support gold** **show tech-support ha** **show tech-support platform**
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	**show environment** **show logging last 200** **show module** **show version**
Inventory	Inventory status that is provided whenever a unit is cold booted or when FRUs are inserted or removed. This alert is considered a noncritical event, and the information is used for status and entitlement.	**show inventory** **show license usage** **show module** **show sprom all** **show system uptime** **show version**
License	Events related to licensing and license violations.	**show logging last 200**

Alert Group	Description	Executed Commands
Linecard hardware	Events related to standard or intelligent switching modules.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
Supervisor hardware	Events related to supervisor modules.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
Syslog port group	Events generated by the syslog PORT facility.	**show license usage** **show logging last 200**

Alert Group	Description	Executed Commands
System	Events generated by failure of a software system that is critical to unit operation.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform**
Test	User-generated test message.	**show module** **show version**

Smart Call Home maps the syslog severity level to the corresponding Smart Call Home severity level for syslog port group messages.

You can customize predefined alert groups to execute additional CLI **show** commands when specific events occur and send that **show** output with the Smart Call Home message.

You can add **show** commands only to full text and XML destination profiles. Short text destination profiles do not support additional **show** commands because they only allow 128 bytes of text.

Smart Call Home Message Levels

Smart Call Home allows you to filter messages based on their level of urgency. You can associate each predefined or user-defined destination profile with a Smart Call Home threshold from 0 (least urgent) to 9 (most urgent). The default is 0 (all messages are sent).

Syslog severity levels are mapped to the Smart Call Home message level.

Note Smart Call Home does not change the syslog message level in the message text.

The following table lists each Smart Call Home message level keyword and the corresponding syslog level for the syslog port alert group.

Table 7: Severity and Syslog Level Mapping

Smart Call Home Level	Keyword	Syslog Level	Description
9	Catastrophic	N/A	Network-wide catastrophic failure.

Smart Call Home Level	Keyword	Syslog Level	Description
8	Disaster	N/A	Significant network impact.
7	Fatal	Emergency (0)	System is unusable.
6	Critical	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	Major	Critical (2)	Major conditions.
4	Minor	Error (3)	Minor conditions.
3	Warning	Warning (4)	Warning conditions.
2	Notification	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	Normal	Information (6)	Normal event signifying return to normal state.
0	Debugging	Debug (7)	Debugging messages.

Obtaining Smart Call Home

If you have a service contract directly with Cisco, you can register for the Smart Call Home service. Smart Call Home analyzes Smart Call Home messages and provides background information and recommendations. For known issues, particularly online diagnostics failures, Automatic Service Requests are generated with the Cisco TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time diagnostic alerts.
- Analysis of Smart Call Home messages and, if needed, Automatic Service Request generation, routed to the correct TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device, through an HTTP proxy server, or a downloadable Transport Gateway (TG). You can use a TG aggregation point to support multiple devices or in cases where security dictates that your devices may not be connected directly to the Internet.
- Web-based access to Smart Call Home messages and recommendations, inventory, and configuration information for all Smart Call Home devices. This feature provides access to associated field notices, security advisories, and end-of-life information.

You need the following information to register:

- The SMARTnet contract number for your device
- Your email address
- Your Cisco.com ID

For more information about Smart Call Home, see the following Smart Call Home page: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

Database Merge Guidelines

When you merge two Smart Call Home databases, the following guidelines apply:

- The merged database contains the following information:
  - A superset of all the destination profiles from the merging devices.
  - The destination profile email addresses and alert groups.
  - Other configuration information (for example, message throttling, or periodic inventory) present in the managing device.
- Destination profile names cannot be duplicated within the merging devices—even though the configurations are different, the names cannot be duplicated. If a profile name is duplicated, one of the duplicate profiles must first be deleted or the merger fails.

High Availability

Both stateful and stateless restarts are supported for Smart Call Home.

Virtualization Support

One instance of Smart Call Home is supported. You can register your contact information at the Smart Call Home web site at the following URL: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

You can test Smart Call Home using the **callhome send** and **callhome test** commands.

Smart Call Home is virtual routing and forwarding (VRF) aware. You can configure Smart Call Home to use a particular VRF to reach the Smart Call Home SMTP server.

Licensing Requirements for Smart Call Home

Product	License Requirement
Cisco NX-OS	Smart Call Home requires no license. Any feature not included in a license package is bundled wit image and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS lice see the Cisco NX-OS Licensing Guide.

Prerequisites for Smart Call Home

Smart Call Home has the following prerequisites:

- To send messages to an email address, you must first configure an email server. To send messages using HTTP, you must have access to an HTTPS server and have a valid certificate installed on the Cisco Nexus device.

- Your device must have IP connectivity to an email server or HTTPS server.
- You must first configure the contact name (SNMP server contact), phone, and street address information. This step is required to determine the origin of messages received.
- If you use Smart Call Home, you need an active service contract for the device that you are configuring.

Guidelines and Limitations for Smart Call Home

Smart Call Home has the following configuration guidelines and limitations:

- If there is no IP connectivity or if the interface in the virtual routing and forwarding (VRF) instance to the profile destination is down, the device cannot send Smart Call Home messages.
- Smart Call Home operates with any SMTP server.
- You can configure up to five SMTP servers for Smart Call Home.
- Link up/down syslog messages do not trigger Smart Call Home messages or alert notifications.

Default Settings for Smart Call Home

This table lists the default settings for Smart Call Home parameters.

Table 8: Default Smart Call Home Parameters

Parameters	Default
Destination message size for a message sent in full text format	2,500,000
Destination message size for a message sent in XML format	2,500,000
Destination message size for a message sent in short text format	4000
SMTP server port number if no port is specified	25
SMTP server priority if no priority is specified	50
Alert group association with profile	All for full-text-destination and short-text-destination profiles. The cisco-tac alert group for the CiscoTAC-1 destination profile.
Format type	XML
Smart Call Home message level	0 (zero)
HTTP proxy server use	Disabled and no proxy server configured

Configuring Smart Call Home

Note Be aware that the Cisco NX-OS commands may differ from the Cisco IOS commands.

We recommend that you complete the Smart Call Home configuration procedures in the following sequence:

1. Configuring Contact Information, on page 51
2. Creating a Destination Profile, on page 53
3. Associating an Alert Group with a Destination Profile, on page 56
4. (Optional) Adding Show Commands to an Alert Group, on page 57
5. Enabling or Disabling Smart Call Home, on page 63
6. (Optional) Testing the Smart Call Home Configuration, on page 64

Configuring Contact Information

You must configure the email, phone, and street address information for Smart Call Home. You can optionally configure the contract ID, customer ID, site ID, and switch priority information.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server contact** *sys-contact* **Example:** `switch(config)# snmp-server contact personname@companyname.com`	Configures the SNMP sysContact.

Step 3	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 4	**email-contact** *email-address* **Example:** `switch(config-callhome)# email-contact admin@Mycompany.com`	Configures the email address for the person primarily responsible for the device. The *email-address* can be up to 255 alphanumeric characters in email address format. **Note** You can use any valid email address. The address cannot contain spaces.

	Command or Action	Purpose
Step 5	**phone-contact** *international-phone-number* **Example:** `switch(config-callhome)# phone-contact +1-800-123-4567`	Configures the phone number in international phone number format for the person primarily responsible for the device. The *international-phone-number* can be up to 17 alphanumeric characters and must be in international phone number format. **Note** The phone number cannot contain spaces. Use the plus (+) prefix before the number.
Step 6	**streetaddress** *address* **Example:** `switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere`	Configures the street address as an alphanumeric string with white spaces for the person primarily responsible for the device. The *address* can be up to 255 alphanumeric characters. Spaces are accepted.
Step 7	(Optional) **contract-id** *contract-number* **Example:** `switch(config-callhome)# contract-id Contract5678`	Configures the contract number for this device from the service agreement. The *contract-number* can be up to 255 alphanumeric characters in free format.
Step 8	(Optional) **customer-id** *customer-number* **Example:** `switch(config-callhome)# customer-id Customer123456`	Configures the customer number for this device from the service agreement. The *customer-number* can be up to 255 alphanumeric characters in free format.
Step 9	(Optional) **site-id** *site-number* **Example:** `switch(config-callhome)# site-id Site1`	Configures the site number for this device. The *site-number* can be up to 255 alphanumeric characters in free format.
Step 10	(Optional) **switch-priority** *number* **Example:** `switch(config-callhome)# switch-priority 3`	Configures the switch priority for this device. The range is from 0 to 7, with 0 being the highest priority and 7 the lowest. The default is 7.
Step 11	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 12	(Optional) **show callhome** **Example:** `switch(config-callhome)# show callhome`	Displays a summary of the Smart Call Home configuration.
Step 13	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

What to do next

Create a destination profile.

Creating a Destination Profile

You can create a user-defined destination profile and configure its message format.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**destination-profile** *name*
	Example:
	`switch(config-callhome)#`
	`destination-profile Noc101`
		Creates a new destination profile. The name can be any alphanumeric string up to 31 characters.

Step 4	**destination-profile** *name* **format** {**XML** | **full-txt** | **short-txt**}
	Example:
	`switch(config-callhome)#`
	`destination-profile Noc101 format full-txt`
		Sets the message format for the profile. The name can be any alphanumeric string up to 31 characters.

Step 5	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 6	(Optional) **show callhome destination-profile** [**profile** *name*] **Example:** `switch(config-callhome)# show callhome destination-profile profile Noc101`	Displays information about one or more destination profiles.

	Command or Action	Purpose
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Associate one or more alert groups with a destination profile.

Modifying a Destination Profile

You can modify the following attributes for a predefined or user-defined destination profile:

- Destination email address—The actual address, pertinent to the transport mechanism, to which the alert should be sent.
- Destination URL—The HTTP or HTTPS URL that defines where alerts should be sent.
- Transport method—The email or HTTP transport that determines which type of destination addresses are used.
- Message formatting—The message format used for sending the alert (full text, short text, or XML).
- Message level—The Smart Call Home message severity level for this destination profile.
- Message size—The allowed length of a Smart Call Home message sent to the email addresses in this destination profile.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **email-addr** *address* **Example:** `switch(config-callhome)# destination-profile full-txt-destination email-addr person@place.com`	Configures an email address for a user-defined or predefined destination profile. You can configure up to 50 email addresses in a destination profile.

	Command or Action	Purpose
Step 4	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **http** *address* **Example:** `switch(config-callhome)# destination-profile CiscoTAC-1 http https://tools.cisco.com/its/service/oddce/services/DDCEService`	Configures an HTTP or HTTPS URL for a user-defined or predefined destination profile. The URL can be up to 255 characters.
Step 5	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **transport-method** {**email** | **http**} **Example:** `switch(config-callhome)# destination-profile CiscoTAC-1 transport-method http`	Configures an email or HTTP transport method for a user-defined or predefined destination profile. The type of transport method that you choose determines the configured destination addresses of that type.
Step 6	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-level** *number* **Example:** `switch(config-callhome)# destination-profile full-txt-destination message-level 5`	Configures the Smart Call Home message severity level for this destination profile. Cisco NX-OS sends only alerts that have a matching or higher Smart Call Home severity level to destinations in this profile. The range is from 0 to 9, where 9 is the highest severity level.
Step 7	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-size** *number* **Example:** `switch(config-callhome)# destination-profile full-txt-destination message-size 100000`	Configures the maximum message size for this destination profile. The range is from 0 to 5000000. The default is 2500000.
Step 8	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 9	(Optional) **show callhome destination-profile** [**profile** *name*] **Example:** `switch(config-callhome)# show callhome destination-profile profile full-text-destination`	Displays information about one or more destination profiles.
Step 10	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

What to do next

Associate one or more alert groups with a destination profile.

Associating an Alert Group with a Destination Profile

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **alert-group** {**All** | **Cisco-TAC** | **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test**} **Example:** `switch(config-callhome)# destination-profile Noc101 alert-group All`	Associates an alert group with this destination profile. Use the **All** keyword to associate all alert groups with the destination profile.
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 5	(Optional) **show callhome destination-profile** [**profile** *name*] **Example:** `switch(config-callhome)# show callhome destination-profile profile Noc101`	Displays information about one or more destination profiles.
Step 6	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

What to do next

Optionally add **show** commands to an alert group and then configure the SMTP email server.

Adding Show Commands to an Alert Group

You can assign a maximum of five user-defined CLI **show** commands to an alert group.

Note You cannot add user-defined CLI **show** commands to the CiscoTAC-1 destination profile.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**alert-group** {**Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test**} **user-def-cmd** *show-cmd* **Example:** `switch(config-callhome)# alert-group Configuration user-def-cmd show ip route`	Adds the **show** command output to any Smart Call Home messages sent for this alert group. Only valid **show** commands are accepted.
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 5	(Optional) **show callhome user-def-cmds** **Example:** `switch(config-callhome)# show callhome user-def-cmds`	Displays information about all user-defined **show** commands added to alert groups.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Configure Smart Call Home to connect to the SMTP email server.

Configuring the Email Server

You must configure the SMTP server address for the Smart Call Home functionality to work. You can also configure the from and reply-to email addresses.

You can configure up to five SMTP servers for Smart Call Home. The servers are tried based on their priority. The highest priority server is tried first. If the message fails to be sent, the next server in the list is tried until the limit is exhausted. If two servers have equal priority, the one that was configured earlier is tried first.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**transport email mail-server** *ip-address* [**port** *number*] [**priority** *number*] [**use-vrf** *vrf-name*] **Example:** `switch(config-callhome)# transport email mail-server 192.0.2.1 use-vrf Red`	Configures the SMTP server as the domain name server (DNS) name, IPv4 address, or IPv6 address. Optionally configures the port number. The port range is from 1 to 65535. The default port number is 25. Also optionally configures the priority of the SMTP server. The priority range is from 1 to 100, with 1 being the highest priority and 100 the lowest. If you do not specify a priority, the default value of 50 is used. Also optionally configures the VRF to use when communicating with this SMTP server. The VRF specified is not used to send messages using HTTP.

	Command or Action	Purpose
Step 4	(Optional) **transport email from** *email-address* **Example:** `switch(config-callhome)# transport email from person@company.com`	Configures the email from field for Smart Call Home messages.
Step 5	(Optional) **transport email reply-to** *email-address* **Example:** `switch(config-callhome)# transport email reply-to person@company.com`	Configures the email reply-to field for Smart Call Home messages.
Step 6	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 7	(Optional) **show callhome transport** **Example:** `switch(config-callhome)# show callhome transport`	Displays the transport-related configuration for Smart Call Home.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Optionally use VRFs to send Smart Call Home messages over HTTP.

Configuring VRFs To Send Messages Using HTTP

You can use VRFs to send Smart Call Home messages over HTTP. If HTTP VRFs are not configured, the default VRF is used to transport messages over HTTP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

	Command or Action	Purpose
Step 3	**transport http use-vrf** *vrf-name* **Example:** `switch(config-callhome)# transport http use-vrf Blue`	Configures the VRF used to send email and other Smart Call Home messages over HTTP.
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 5	(Optional) **show callhome** **Example:** `switch(config-callhome)# show callhome`	Displays information about Smart Call Home.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Optionally configure Smart Call Home to send HTTP messages through an HTTP proxy server.

Configuring an HTTP Proxy Server

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**transport http proxy server** *ip-address* [**port** *number*] **Example:** `switch(config-callhome)# transport http proxy server 192.0.2.1`	Configures the HTTP proxy server domain name server (DNS) name, IPv4 address, or IPv6 address. Optionally configures the port number. The port range is from 1 to 65535. The default port number is 8080.

	Command or Action	Purpose
Step 4	**transport http proxy enable** **Example:** `switch(config-callhome)# transport http proxy enable`	Enables Smart Call Home to send all HTTP messages through the HTTP proxy server. **Note** You can execute this command only after the proxy server address has been configured. **Note** The VRF used for transporting messages through the proxy server is the same as that configured using the **transport http use-vrf** command.
Step 5	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 6	(Optional) **show callhome transport** **Example:** `switch(config-callhome)# show callhome transport`	Displays the transport-related configuration for Smart Call Home.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Optionally configure your device to periodically send inventory notifications.

Configuring Periodic Inventory Notifications

You can configure the device to periodically send a message with an inventory of all software services currently enabled and running on the device along with hardware inventory information. The device generates two Smart Call Home notifications: periodic configuration messages and periodic inventory messages.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome** **Example:**	Enters Smart Call Home configuration mode.

	Command or Action	Purpose
	`switch(config)# callhome` `switch(config-callhome)#`	
Step 3	**periodic-inventory notification** [**interval** *days*] [**timeofday** *time*] **Example:** `switch(config-callhome)# periodic-inventory notification interval 20`	Configures periodic inventory messages. The interval range is from 1 to 30 days, and the default is 7 days. The *time* argument is in HH:MM format. It defines at what time of the day every *X* days an update is sent (where *X* is the update interval).
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 5	(Optional) **show callhome** **Example:** `switch(config-callhome)# show callhome`	Displays information about Smart Call Home.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Optionally disable duplicate message throttling.

Disabling Duplicate Message Throttling

You can limit the number of duplicate messages received for the same event. By default, the device limits the number of duplicate messages received for the same event. If the number of duplicate messages sent exceeds 30 messages within a 2-hour time frame, the device discards further messages for that alert type.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

	Command or Action	Purpose
Step 3	**no duplicate-message throttle** **Example:** `switch(config-callhome)# no duplicate-message throttle`	Disables duplicate message throttling for Smart Call Home. Duplicate message throttling is enabled by default.
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Enable Smart Call Home.

Enabling or Disabling Smart Call Home

Once you have configured the contact information, you can enable the Smart Call Home function.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	[**no**] **enable** **Example:** `switch(config-callhome)# enable`	Enables or disables Smart Call Home. Smart Call Home is disabled by default.
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the Smart Call Home configuration commands.
Step 5	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

What to do next

Optionally generate a test message.

Testing the Smart Call Home Configuration

You can generate a test message to test your Smart Call Home communications.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**callhome**
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Enters Smart Call Home configuration mode.

Step 3	**callhome send** [**configuration** | **diagnostic**] **Example:** `switch(config-callhome)# callhome send diagnostic`	Sends the specified Smart Call Home test message to all configured destinations.
Step 4	**callhome test** **Example:** `switch(config-callhome)# callhome test`	Sends a test message to all configured destinations.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Smart Call Home Configuration

To display Smart Call Home configuration information, perform one of the following tasks:

Command	Purpose
show callhome	Displays the Smart Call Home configuration.

Command	Purpose
show callhome destination-profile *name*	Displays one or more Smart Call Home destination profiles.
show callhome transport	Displays the transport-related configuration for Smart Call Home.
show callhome user-def-cmds	Displays CLI commands added to any alert groups.
show running-config callhome [**all**]	Displays the running configuration for Smart Call Home.
show startup-config callhome	Displays the startup configuration for Smart Call Home.
show tech-support callhome	Displays the technical support output for Smart Call Home.

Configuration Examples for Smart Call Home

This example shows how to create a destination profile called Noc101, associate the Configuration alert group to that profile, configure contact and email information, and specify the VRF used to send Smart Call Home messages over HTTP:

```
configure terminal
snmp-server contact person@company.com
callhome
distribute
email-contact admin@Mycompany.com
phone-contact +1-800-123-4567
streetaddress 123 Anystreet st. Anytown,AnyWhere
destination-profile Noc101 format full-txt
destination-profile full-text-destination email-addr person@company.com
destination-profile full-text-destination message-level 5
destination-profile Noc101 alert-group Configuration
alert-group Configuration user-def-cmd show ip route
transport email mail-server 192.0.2.10 priority 1
transport http use-vrf Blue
enable
commit
```

This example shows how to configure multiple SMTP servers for Smart Call Home messages:

```
configure terminal
callhome
transport email mail-server 192.0.2.10 priority 4
transport email mail-server 172.21.34.193
transport email smtp-server 10.1.1.174
transport email mail-server 64.72.101.213 priority 60
transport email from person@company.com
transport email reply-to person@company.com
commit
```

Based on the configuration above, the SMTP servers would be tried in this order:

10.1.1.174 (priority 0)

192.0.2.10 (priority 4)

172.21.34.193 (priority 50, which is the default)

64.72.101.213 (priority 60)

Note The **transport email smtp-server** command has a priority of 0, which is the highest. The server specified by this command is tried first followed by the servers specified by the **transport email mail-server** commands in order of priority.

This example shows how to configure Smart Call Home to send HTTP messages through an HTTP proxy server:

```
configure terminal
callhome
transport http proxy server 10.10.10.1 port 4
transport http proxy enable
commit
```

Additional References

Event Triggers

The following table lists the event triggers and their Smart Call Home message severity levels.

Alert Group	Event Name	Description	Smart Call Home Severity Level
Configuration	PERIODIC_CONFIGURATION	Periodic configuration update message.	2
Diagnostic	DIAGNOSTIC_MAJOR_ALERT	GOLD generated a major alert.	7
	DIAGNOSTIC_MINOR_ALERT	GOLD generated a minor alert.	4
	DIAGNOSTIC_NORMAL_ALERT	Smart Call Home generated a normal diagnostic alert.	2
Environmental and CISCO_TAC	FAN_FAILURE	Cooling fan has failed.	5
	POWER_SUPPLY_ALERT	Power supply warning has occurred.	6
	POWER_SUPPLY_FAILURE	Power supply has failed.	6
	POWER_SUPPLY_SHUTDOWN	Power supply has shut down.	6
	TEMPERATURE_ALARM	Thermal sensor going bad.	6
	TEMPERATURE_MAJOR_ALARM	Thermal sensor indicates temperature has reached operating major threshold.	6
	TEMPERATURE_MINOR_ALARM	Thermal sensor indicates temperature has reached operating minor threshold.	4

Alert Group	Event Name	Description	Smart Call Home Severity Level
Inventory and CISCO_TAC	COLD_BOOT	Switch is powered up and reset to a cold boot sequence.	2
	HARDWARE_INSERTION	New piece of hardware has been inserted into the chassis.	2
	HARDWARE_REMOVAL	Hardware has been removed from the chassis.	2
	PERIODIC_INVENTORY	Periodic inventory message has been generated.	2
License	LICENSE_VIOLATION	Feature in use is not licensed and is turned off after grace period expiration.	6
Line module Hardware and CISCO_TAC	LINEmodule_FAILURE	Module operation has failed.	7
Supervisor Hardware and CISCO_TAC	SUP_FAILURE	Supervisor module operation has failed.	7
Syslog-group-port	PORT_FAILURE	syslog message that corresponds to the port facility has been generated.	6
	SYSLOG_ALERT	syslog alert message has been generated. **Note** Link up/down syslog messages do not trigger Smart Call Home messages or alert notifications.	5
System and CISCO_TAC	SW_CRASH	Software process has failed with a stateless restart, indicating an interruption of a service. Messages are sent for process crashes on supervisor modules.	5
	SW_SYSTEM_INCONSISTENT	Inconsistency has been detected in software or file system.	5
Test and CISCO_TAC	TEST	User generated test has occurred.	2

Message Formats

Smart Call Home supports the following message formats:

Short Text Message Format

The following table describes the short text formatting option for all message types.

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to system message

Common Event Message Fields

The following table describes the first set of common event message fields for full text or XML messages.

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Timestamp	Date and time stamp of event in ISO time notation: YYYY-MM-DD HH:MM:SS GMT+HH:MM.	/aml/header/time
Message name	Name of message.	/aml/header/name
Message type	Name of message type, such as reactive or proactive.	/aml/header/type
Message group	Name of alert group, such as syslog.	/aml/header/group
Severity level	Severity level of message.	/aml/header/level
Source ID	Product type for routing, such as the Cisco Nexus 9000 Series switch.	/aml/header/source

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Device ID	Unique device identifier (UDI) for the end device that generated the message. This field should be empty if the message is nonspecific to a device. The format is *type@Sid@serial*. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is N9K-C9508@C@12345678.	/aml/ header/deviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header/customerID
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header /contractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	/aml/ header/siteId
Server ID	If the message is generated from the device, this ID is the unique device identifier (UDI) of the device. The format is *type@Sid@serial*. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is N9K-C9508@C@12345678.	/aml/header/serverId
Message description	Short text that describes the error.	/aml/body/msgDesc

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Device name	Node that experienced the event (hostname of the device).	/aml/body/sysName
Contact name	Name of person to contact for issues associated with the node that experienced the event.	/aml/body/sysContact
Contact email	Email address of person identified as the contact for this unit.	/aml/body/sysContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	/aml/body/sysContactPhone Number
Street address	Optional field that contains the street address for RMA part shipments associated with this unit.	/aml/body/sysStreetAddress
Model name	Model name of the device (the specific model as part of a product family name).	/aml/body/chassis/name
Serial number	Chassis serial number of the unit.	/aml/body/chassis/serialNo
Chassis part number	Top assembly number of the chassis.	/aml/body/chassis/partNo

Alert Group Message Fields

The following table describes the fields specific to alert group messages for full text and XML. These fields may be repeated if multiple CLI commands are executed for an alert group.

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Command output name	Exact name of the issued CLI command.	/aml/attachments/attachment/name
Attachment type	Specific command output.	/aml/attachments/attachment/type
MIME type	Either plain text or encoding type.	/aml/attachments/attachment/mime
Command output text	Output of command automatically executed.	/aml/attachments/attachment/atdata

Fields for Reactive and Proactive Event Messages

The following table describes the reactive and proactive event message format for full text or XML messages.

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Affected FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
Affected FRU serial number	Serial number of the affected FRU.	/aml/body/fru/serialNo
Affected FRU part number	Part number of the affected FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU that is generating the event message.	/aml/body/fru/slot
FRU hardware version	Hardware version of the affected FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the affected FRU.	/aml/body/fru/swVersion

Fields for Inventory Event Messages

The following table describes the inventory event message format for full text or XML messages.

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of the chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
FRU s/n	Serial number of the FRU.	/aml/body/fru/serialNo
FRU part number	Part number of the FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU.	/aml/body/fru/slot
FRU hardware version	Hardware version of the FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the FRU.	/aml/body/fru/swVersion

Fields for User-Generated Test Messages

The following table describes the user-generated test message format for full text or XML.

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Sample Syslog Alert Notification in Full-Text Format

This sample shows the full-text format for a syslog port alert-group notification:

```
Severity Level:5
Series:Nexus9000
Switch Priority:0
Device Id:N9K-C9508@C@TXX12345678
Server Id:N9K-C9508C@TXX12345678
Time of Event:2013-05-17 16:31:33 GMT+0000 Message Name:
Message Type:syslog
System Name:dc3-test
Contact Name:Jay Tester
Contact Email:contact@example.com
Contact Phone:+91-80-1234-5678
Street Address:#1 Any Street
Event Description:SYSLOG_ALERT 2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error
 (0x20) while communicating with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP
 (for:RID_PORT: Ethernet3/1)

syslog_facility:ETHPORT
start chassis information:
Affected Chassis:N9K-C9508
Affected Chassis Serial Number:TXX12345678 Affected Chassis Hardware Version:0.405 Affected
 Chassis Software Version:6.1(2) Affected Chassis Part No:11-11111-11 end chassis information:
start attachment
   name:show logging logfile | tail -n 200
   type:text
   data:
   2013 May 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages) cleared
 by user
   2013 May 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
   2013 May 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
   2013 May 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16: Invalid
 argument: - sshd[14484]
   2013 May 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
   2013 May 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: "System Manager
(gsync controller)" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
   2013 May 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
   2013 May 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 3504) hasn't
 caught signal 9 (no core).
   2013 May 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
   2013 May 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23210)
hasn't caught signal 9 (no core).
   2013 May 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
   2013 May 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23294)
hasn't caught signal 9 (no core).
   2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
   2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is becoming
 active.
   2013 May 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed -
 device_test
   2013 May 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR: netstack [4336] Unrecognized
 message from MRIB. Major type 1807
```

```
   2013 May 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN
   2013 May 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10 - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]
  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined - ntpd[19045]

   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
   2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
   2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
recovery failed (0)
   2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
recovery failed (0)
   2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
   2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
   2013 May 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
   2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
   2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
   2013 May 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP
   2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
   2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
   2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present but
all AC inputs are not connected, ac-redundancy might be affected
   2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present but
all AC inputs are not connected, ac-redundancy might be affected
   2013 May 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
  2013 May 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
  2013 May 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
  2013 May 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
  2013 May 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
   2013 May 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
  2013 May 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 4820) hasn't
 caught signal 9 (no core).
   2013 May 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24239)
hasn't caught signal 9 (no core).
   2013 May 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24401)
hasn't caught signal 9 (no core).
   2013 May 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
   2013 May 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24407)
hasn't caught signal 9 (no core).
  2013 May 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
  2013 May 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
```

```
  2013 May 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command <more>
 return code <14>
  2013 May 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR: netstack [4336] (null)
  2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1)
 end attachment start attachment
  type:text
  data:

  dc3-test interfaces:
       Ethernet3/1    Ethernet3/2    Ethernet3/3
       Ethernet3/4    Ethernet3/5    Ethernet3/6
       Ethernet3/7    Ethernet3/8    Ethernet3/9
       Ethernet3/10   Ethernet3/11   Ethernet3/12
       Ethernet3/13   Ethernet3/14   Ethernet3/15
       Ethernet3/16   Ethernet3/17   Ethernet3/18
       Ethernet3/19   Ethernet3/20   Ethernet3/21
       Ethernet3/22   Ethernet3/23   Ethernet3/24
       Ethernet3/25   Ethernet3/29   Ethernet3/30
       Ethernet3/31   Ethernet3/32   Ethernet3/33
       Ethernet3/34   Ethernet3/35   Ethernet3/36
       Ethernet3/37   Ethernet3/38   Ethernet3/39
       Ethernet3/40   Ethernet3/41   Ethernet3/42
       Ethernet3/43   Ethernet3/44   Ethernet3/45
       Ethernet3/46   Ethernet3/47   Ethernet3/48
end attachment
start attachment
   type:text
   data:
end attachment
start attachment
   name:show license usage
   type:text
   data:
   Feature Ins Lic Status Expiry Date Comments
             Count
   --------------------------------------------------------------------------------
   LAN_ENTERPRISE_SERVICES_PKG Yes - Unused Never -
   --------------------------------------------------------------------------------
end attachment
```

Sample Syslog Alert Notification in XML Format

This sample shows the XML format for a syslog port alert-group notification:

```
<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.cisco.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>1004:TXX12345678:478F82E6</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
```

```
<aml-block:Type>http://www.cisco.com/2005/05/callhome/syslog</aml-block:Type>
<aml-block:CreationDate>2013-05-17 16:31:33 GMT+0000</aml-block:CreationDate>
<aml-block:Builder> <aml-block:Name>DC3</aml-block:Name>
<aml-block:Version>4.1</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>1005:TXX12345678:478F82E6</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>5</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.cisco.com/2005/05/callhome" version="1.0">
<ch:EventTime>2013-05-17 16:31:33 GMT+0000</ch:EventTime> <ch:MessageDescription>SYSLOG_ALERT
 2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1)
 </ch:MessageDescription>
<ch:Event> <ch:Type>syslog</ch:Type> <ch:SubType></ch:SubType> <ch:Brand>Cisco</ch:Brand>
<ch:Series>Nexus9000</ch:Series> </ch:Event> <ch:CustomerData> <ch:UserData>
<ch:Email>contact@example.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:DeviceId>N9K-C9508@C@TXX12345678</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>dc3-test</ch:Name>
<ch:Contact>Jay Tester</ch:Contact> <ch:ContactEmail>contact@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber>+91-80-1234-5678</ch:ContactPhoneNumber>
<ch:StreetAddress>#1, Any Street</ch:StreetAddress> </ch:SystemInfo> </ch:CustomerData>
<ch:Device> <rme:Chassis xmlns:rme="http://www.cisco.com/rme/4.1">
<rme:Model>N9K-C9508</rme:Model>
<rme:HardwareVersion>0.405</rme:HardwareVersion>
<rme:SerialNumber>TXX12345678</rme:SerialNumber>
</rme:Chassis>
</ch:Device>
</ch:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging logfile | tail -n 200</aml-block:Name> <aml-block:Data
encoding="plain">
<![CDATA[2013 May 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
cleared by user
2013 May 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16: Invalid
argument: - sshd[14484]
2013 May 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: \"System Manager (gsync
 controller)\" (PID 12000) has finished with error code SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL
 (12).
2013 May 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message Core
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 3504)
hasn&amp;apos;t caught signal 9 (no core).
2013 May 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message Core
```

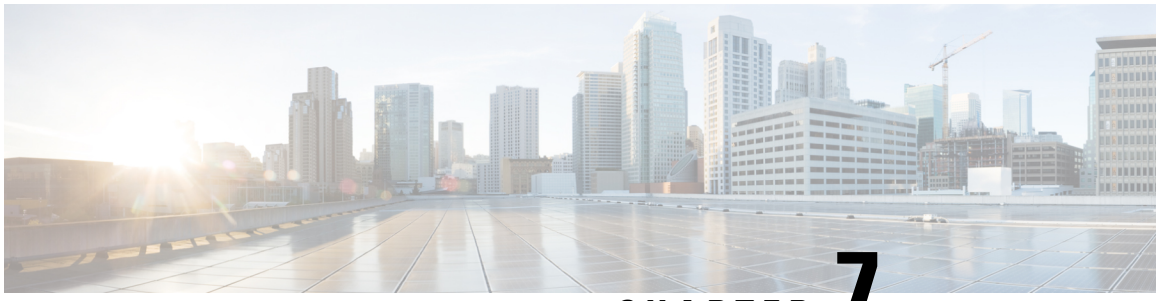
```
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23210)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message Core
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23294)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is becoming
 active (pre-start phase).
2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is becoming
active.
2013 May 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed -
device_test
2013 May 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR: netstack [4336] Unrecognized
message from MRIB. Major type 1807
2013 May 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN
2013 May 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
recovery failed (0)
2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
recovery failed (0)
2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
2013 May 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
2013 May 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP
2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present but all
 AC inputs are not connected, ac-redundancy might be affected
2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present but all
 AC inputs are not connected, ac-redundancy might be affected
2013 May 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
2013 May 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 4820)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24239)
hasn&apos;t caught signal 9 (no core).
```

```
2013 May 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24401)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
2013 May 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
 not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24407)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR: netstack [4336] (null)
2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1)
 ]]> </aml-block:Data> </aml-block:Attachment> <aml-block:Attachment type="inline">
<aml-block:Name> <aml-block:Data encoding="plain"> <![CDATA[
dc3-test interfaces:
   Ethernet3/1     Ethernet3/2     Ethernet3/3
   Ethernet3/4     Ethernet3/5     Ethernet3/6
   Ethernet3/7     Ethernet3/8     Ethernet3/9
   Ethernet3/10    Ethernet3/11    Ethernet3/12
   Ethernet3/13    Ethernet3/14    Ethernet3/15
   Ethernet3/16    Ethernet3/17    Ethernet3/18
   Ethernet3/19    Ethernet3/20    Ethernet3/21
   Ethernet3/22    Ethernet3/23    Ethernet3/24
   Ethernet3/25    Ethernet3/26    Ethernet3/27
   Ethernet3/28    Ethernet3/29    Ethernet3/30
   Ethernet3/31    Ethernet3/32    Ethernet3/33
   Ethernet3/34    Ethernet3/35    Ethernet3/36
   Ethernet3/37    Ethernet3/38    Ethernet3/39
   Ethernet3/40    Ethernet3/41    Ethernet3/42
   Ethernet3/43    Ethernet3/44    Ethernet3/45
   Ethernet3/46    Ethernet3/47    Ethernet3/48


]]>
</aml-block:Data>
</aml-block:Attachment>
<aml-block:Attachment type="inline">
<aml-block:Name> <aml-block:Data encoding="plain"> <!> </aml-block:Data>
</aml-block:Attachment> <aml-block:Attachment type="inline"> <aml-block:Name>show license
usage</aml-block:Name> <aml-block:Data encoding="plain">
<![CDATA[Feature Ins Lic Status Expiry Date Comments
                    Count
--------------------------------------------------------------------------------
LAN_ENTERPRISE_SERVICES_PKG Yes - Unused Never -
--------------------------------------------------------------------------------
]]>
</aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

MIBs

MIBs	MIBs Link
MIBs related to Smart Call Home	To locate and download supported MIBs, go to the following ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 7

Configuring Rollback

This chapter describes how to configure rollback on Cisco NX-OS devices.

This chapter contains the following sections:

About Rollbacks

A rollback allows you to take a snapshot, or user checkpoint, of the Cisco NX-OS configuration and then reapply that configuration to your device at any point without having to reload the device. A rollback allows any authorized administrator to apply this checkpoint configuration without requiring expert knowledge of the features configured in the checkpoint.

Cisco NX-OS automatically creates system checkpoints. You can use either a user or system checkpoint to perform a rollback.

You can create a checkpoint copy of the current running configuration at any time. Cisco NX-OS saves this checkpoint as an ASCII file which you can use to roll back the running configuration to the checkpoint configuration at a future time. You can create multiple checkpoints to save different versions of your running configuration.

When you roll back the running configuration, you can trigger the following rollback types:

- atomic—Implement a rollback only if no errors occur.
- best-effort—Implement a rollback and skip any errors.
- stop-at-first-failure—Implement a rollback that stops if an error occurs.

The default rollback type is atomic.

When you are ready to roll back to a checkpoint configuration, you can view the changes that will be applied to your current running configuration before committing to the rollback operation. If an error occurs during the rollback operation, you can choose to cancel the operation, or ignore the error and proceed with the rollback.

If you cancel the operation, Cisco NX-OS provides a list of changes already applied before the error occurred. You need to clean up these changes manually.

Automatically Generated System Checkpoints

The Cisco NX-OS software automatically generates system checkpoints to help you avoid a loss of configuration information. System checkpoints are generated by the following events:

- Disabling an enabled feature with the **no feature** command
- Removing an instance of a Layer 3 protocol, such as with the **no router bgp** command or the **no ip pim sparse-mode** command
- License expiration of a feature

If one of these events causes system configuration changes, the feature software creates a system checkpoint that you can use to roll back to the previous system configuration. The system generated checkpoint filenames begin with "system-" and include the feature name. For example, the first time that you disable the EIGRP feature, the system creates the checkpoint named system-fm-__inst_1__eigrp.

High Availability

Whenever a checkpoint is created using the checkpoint or checkpoint checkpoint_name commands, the checkpoint is synchronized to the standby unit.

A rollback remembers the states of the checkpoint operation, so if the checkpoint operation is interrupted and the system is left in an inconsistent state, a rollback can complete the checkpoint operation (synchronize the checkpoint with the standby unit) before proceeding with the rollback operation.

Your checkpoint files are still available after a process restart or supervisor switchover. Even if there is an interruption during the process restart or supervisor switchover, the checkpoint will complete successfully before proceeding with the operation. In a supervisor switchover, the checkpoint is completed on the new active unit.

If a process restart or supervisor switchover occurs during a rollback operation, after the restart or switchover completes, the rollback will resume from its previous state and complete successfully.

Virtualization Support

Cisco NX-OS creates a checkpoint of the running configuration. You can create different checkpoint copies.

Prerequisites for Rollbacks

To configure rollback, you must have network-admin user privileges.

Guidelines and Limitations for Rollbacks

Rollbacks have the following configuration guidelines and limitations:

- You can create up to ten checkpoint copies.

- Your checkpoint filenames must be 80 characters or less.
- You cannot start a checkpoint filename with the word *system*.
- You can start a checkpoint filename with the word *auto*.
- You can name a checkpoint file *summary* or any abbreviation of the word *summary*.
- Only one user can perform a checkpoint, rollback, or copy the running configuration to the startup configuration at the same time.
- After the system executes the **write erase** or **reload** command, checkpoints are deleted. You can use the **clear checkpoint database** command to clear out all checkpoint files.
- Although a rollback is not supported for checkpoints across software versions, users can perform a rollback at their own discretion and can use the best-effort mode to recover from errors.
- When checkpoints are created on bootflash, differences with the running-system configuration cannot be performed before performing the rollback, and the system reports "No Changes."
- Checkpoints created using the **checkpoint** and **checkpoint** *checkpoint_name* commands are present upon a switchover.
- Checkpoints are present upon reload unless a **write-erase** command is issued before a reload.
- A rollback to files on bootflash is supported only on files created using the **checkpoint** *checkpoint_name* command and not on any other type of ASCII file.
- Checkpoint names must be unique. You cannot overwrite previously saved checkpoints with the same name.
- Rollback is not supported in the context of auto configurations. Checkpoints do not store auto configurations. Therefore, after a rollback is performed, the corresponding auto configurations will not be present
- Multiple port VLAN mappings configured on an interface during a rollback operation causes the rollback feature to fail.

Default Settings for Rollbacks

This table lists the default settings for rollback parameters.

Parameters	Default
Rollback type	Atomic

Configuring Rollbacks

Note Be aware that the Cisco NX-OS commands may differ from the Cisco IOS commands.

Creating a Checkpoint

You can create up to ten checkpoints of your configuration.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **checkpoint** {[*cp-name*] [**description** *descr*] | **file** *file-name* } **Example:** `switch# checkpoint stable`	Creates a checkpoint of the running configuration to either a user checkpoint name or a file. The checkpoint name can be any alphanumeric string up to 80 characters but cannot contain spaces. If you do not provide a name, Cisco NX-OS sets the checkpoint name to user-checkpoint-*number* where *number* is from 1 to 10. The description can contain up to 80 alphanumeric characters, including spaces. You can use the **no** form of the **checkpoint** command to remove a checkpoint name. Use the **delete** command to remove a checkpoint file.
Step 2	(Optional) **show checkpoint** *cp-name* [**all**] **Example:** `switch# show checkpoint stable`	Displays the contents of the checkpoint name.

Implementing a Rollback

You can implement a rollback to a checkpoint name or file. Before you implement a rollback, you can view the differences between source and destination checkpoints that reference current or saved configurations.

Note If you make a configuration change during an atomic rollback, the rollback will fail.

Procedure

	Command or Action	Purpose
Step 1	**show diff rollback-patch** {**checkpoint** *src-cp-name* | **running-config** | **startup-config** | **file** *source-file*} {**checkpoint** *dest-cp-name* | **running-config** | **startup-config** | **file** *dest-file*} **Example:** `switch# show diff rollback-patch checkpoint stable running-config`	Displays the differences between the source and destination checkpoint selections.

	Command or Action	Purpose
Step 2	**rollback running-config** {**checkpoint** *cp-name* | **file** *cp-file*} [**atomic** | **best-effort** | **stop-at-first-failure**] **Example:** `switch# rollback running-config checkpoint stable`	Creates a rollback to the specified checkpoint name or file. You can implement the following rollback types: • atomic—Implement a rollback only if no errors occur. • best-effort—Implement a rollback and skip any errors. • stop-at-first-failure—Implement a rollback that stops if an error occurs. The default is atomic. This example shows how to implement a rollback to a user checkpoint name.

Verifying the Rollback Configuration

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

Command	Purpose
show checkpoint *name* [**all**]	Displays the contents of the checkpoint name.
show checkpoint all [**user** | **system**]	Displays the contents of all checkpoints. You can limit the displayed checkpoints to user or system generated checkpoints.
show checkpoint summary [**user** | **system**]	Displays a list of all checkpoints. You can limit the displayed checkpoints to user or system generated checkpoints.
show diff rollback-patch {**checkpoint** *src-cp-name* | **running-config** | **startup-config** | **file** *source-file*} {**checkpoint** *dest-cp-name* | **running-config** | **startup-config** | **file** *dest-file*}	Displays the differences between the source and destination checkpoint selections.
show rollback log [**exec** | **verify**]	Displays the contents of the rollback log.

Use the **clear checkpoint database** command to delete all checkpoint files.

Configuration Example for Rollback

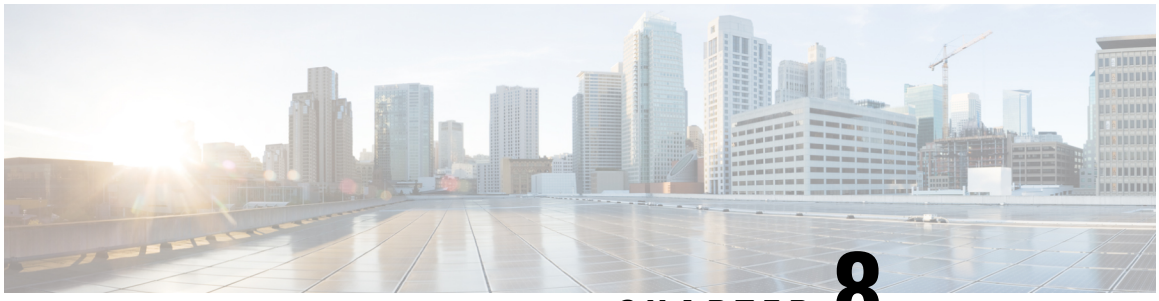
This example shows how to create a checkpoint file and then implements a best-effort rollback to a user checkpoint name:

```
checkpoint stable
rollback running-config checkpoint stable best-effort
```

Additional References

Related Documents

Related Topic	Document Title
Configuration files	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 8

Configuring Session Manager

This chapter describes how to configure Session Manager on Cisco NX-OS devices.

This chapter contains the following sections:

About Session Manager

Session Manager allows you to implement your configuration changes in batch mode. Session Manager works in the following phases:

- Configuration session—Creates a list of commands that you want to implement in Session Manager mode.
- Validation—Provides a basic semantic check on your configuration. Cisco NX-OS returns an error if the semantic check fails on any part of the configuration.
- Verification—Verifies the configuration as a whole, based on the existing hardware and software configuration and resources. Cisco NX-OS returns an error if the configuration does not pass this verification phase.
- Commit—Cisco NX-OS verifies the complete configuration and applies the changes to the device. If a failure occurs, Cisco NX-OS reverts to the original configuration.
- Abort—Discards the configuration changes before implementation.

You can optionally end a configuration session without committing the changes. You can also save a configuration session.

High Availability

Session Manager sessions remain available after a supervisor switchover. Sessions are not persistent across a software reload.

Prerequisites for Session Manager

Make sure that you have the privilege level required to support the Session Manager commands that you plan to use.

Guidelines and Limitations for Session Manager

Session Manager has the following configuration guidelines and limitations:

- Session Manager supports only access control list (ACL) and quality of service (QoS) features.
- You can create up to 32 configuration sessions.
- You can configure a maximum of 20,000 commands across all sessions.
- You cannot simultaneously execute configuration commands in more then one configuration session or configuration terminal mode. Parallel configurations (for example, one configuration session and one configuration terminal) might cause validation or verification failures in the configuration session.
- If an interface reloads while you are configuring that interface in a configuration session, Session Manager may accept the commands even though the interface is not present in the device at that time.

Configuring Session Manager

Note Be aware that the Cisco NX-OS commands might differ from Cisco IOS commands.

Creating a Session

You can create up to 32 configuration sessions.

Procedure

	Command or Action	Purpose

Step 1	**configure session** *name*
	Example:
	`switch# configure session myACLs`
	`switch(config-s)#`
		Creates a configuration session and enters session configuration mode. The name can be any alphanumeric string.
		Displays the contents of the session.

Step 2	(Optional) **show configuration session** [*name*]	Displays the contents of the session.

	Command or Action	Purpose
	Example: `switch(config-s)# show configuration session myACLs`	
Step 3	(Optional) **save** *location* **Example:** `switch(config-s)# save bootflash:sessions/myACLs`	Saves the session to a file. The location can be in bootflash:, slot0:, or volatile:.

Configuring ACLs in a Session

You can configure ACLs within a configuration session.

Procedure

	Command or Action	Purpose

Step 1	**configure session** *name*
	Example:
	`switch# configure session myacls`
	`switch(config-s)#`
		Creates a configuration session and enters session configuration mode. The name can be any alphanumeric string.

Step 2	**ip access-list** *name*
	Example:
	`switch(config-s)# ip access-list acl1`
	`switch(config-s-acl)#`
		Creates an ACL and enters a configuration mode for that ACL.

Step 3	(Optional) **permit** *protocol source destination* **Example:** `switch(config-s-acl)# permit tcp any any`	Adds a permit statement to the ACL.

Step 4	**interface** *interface-type number*
	Example:
	`switch(config-s-acl)# interface ethernet 2/1`
	`switch(config-s-if)#`
		Enters interface configuration mode.

Step 5	**ip access-group** *name* {**in** | **out**} **Example:** `switch(config-s-if)# ip access-group acl1 in`	Specifies the direction of traffic the access group is applied to.
Step 6	(Optional) **show configuration session** [*name*] **Example:** `switch(config-s-if)# show configuration session myacls`	Displays the contents of the session.

Verifying a Session

Use the following command in session mode to verify a session:

Command	Purpose
verify [**verbose**] **Example:** `switch(config-s)# verify`	Verifies the configuration as a whole, based on the existing hardware and software configuration and resources. Cisco NX-OS returns an error if the configuration does not pass this verification.

Committing a Session

Use the following command in session mode to commit a session:

Command	Purpose
commit [**verbose**] **Example:** `switch(config-s)# commit`	Validates the configuration changes made in the current session and applies valid changes to the device. If the validation fails, Cisco NX-OS reverts to the original configuration.

Saving a Session

Use the following command in session mode to save a session:

Command	Purpose
save *location* **Example:** `switch(config-s)# save bootflash:sessions/myACLs`	(Optional) Saves the session to a file. The location can be in bootflash:, slot0:, or volatile:.

Discarding a Session

Use the following command in session mode to discard a session:

Command	Purpose

abort
Example:
`switch(config-s)# abort`
`switch#`
	Discards the configuration session without applying the changes.

Verifying the Session Manager Configuration

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

Command	Purpose
show configuration session [*name*]	Displays the contents of the configuration session.
show configuration session status [*name*]	Displays the status of the configuration session.
show configuration session summary	Displays a summary of all the configuration sessions.

Configuration Example for Session Manager

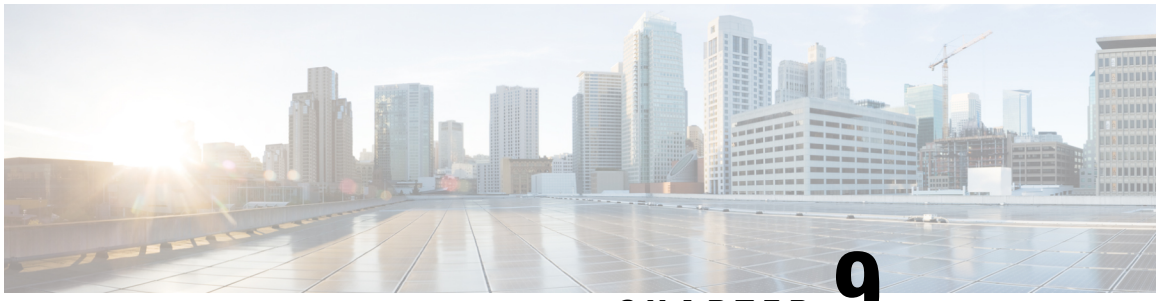
This example shows how to create and commit an ACL configuration using Session Manager:

```
switch# configure session ACL_tcp_in
Config Session started, Session ID is 1
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-s)# ip access-list ACL1
switch(config-s-acl)# permit tcp any any
switch(config)# interface e 7/1
switch(config-if)# ip access-group ACL1 in
switch(config-if)# exit
switch(config)# exit
switch# config session ACL_tcp_in
Config Session started, Session ID is 1
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-s)# verify
Verification Successful
switch(config-s)# commit
Commit Successful
switch#
```

Additional References

Related Documents

Related Topic	Document Title
Configuration files	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 9

Configuring the Scheduler

This chapter describes how to configure the scheduler on Cisco NX-OS devices.

This chapter includes the following sections:

About the Scheduler

The scheduler allows you to define and set a timetable for maintenance activities such as the following:

- Quality of service (QoS) policy changes
- Data backup
- Saving a configuration

Jobs consist of a single command or multiple commands that define routine activities. Jobs can be scheduled one time or at periodic intervals.

The scheduler defines a job and its timetable as follows:

- Job—A routine task or tasks defined as a command list and completed according to a specified schedule.
- Schedule—The timetable for completing a job. You can assign multiple jobs to a schedule. A schedule is defined as either periodic or one-time only:
  - Periodic mode—A recurring interval that continues until you delete the job. You can configure the following types of intervals:
    - Daily—A job is completed once a day.
    - Weekly—A job is completed once a week.
    - Monthly—A job is completed once a month.
    - Delta—A job begins at the specified start time and then at specified intervals (days:hours:minutes).

- One-time mode—A job is completed only once at a specified time.

Remote User Authentication

Before starting a job, the scheduler authenticates the user who created the job. Since user credentials from a remote authentication are not retained long enough to support a scheduled job, you need to locally configure the authentication passwords for users who create jobs. These passwords are part of the scheduler configuration and are not considered a locally configured user.

Before starting the job, the scheduler validates the local password against the password from the remote authentication server.

Logs

The scheduler maintains a log file containing the job output. If the size of the job output is greater than the size of the log file, the output is truncated.

High Availability

Scheduled jobs remain available after a supervisor switchover or a software reload.

Prerequisites for the Scheduler

The scheduler has the following prerequisites:

- You must enable any conditional features before you can configure those features in a job.
- You must have a valid license installed for any licensed features that you want to configure in the job.
- You must have network-admin user privileges to configure a scheduled job.

Guidelines and Limitations for the Scheduler

The scheduler has the following configuration guidelines and limitations:

- The scheduler can fail if it encounters one of the following while performing a job:
  - Verify that you have configured the time. The scheduler does not apply a default timetable. If you create a schedule and assign jobs and do not configure the time, the job is not started.
  - While defining a job, verify that no interactive or disruptive commands (for example, **copy bootflash:** *file* **ftp:** *URI*, **write erase**, and other similar commands) are specified because the job is started and conducted noninteractively.

Default Settings for the Scheduler

This table lists the scheduler default settings.

Parameters	Default
Scheduler state	Disabled
Log file size	16 KB

Configuring the Scheduler

Enabling or Disabling the Scheduler

You can enable the scheduler feature so that you can configure and schedule jobs, or you can disable the scheduler feature after it has been enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature scheduler** **Example:** `switch(config)# feature scheduler`	Enables or disables the scheduler.

Step 3	(Optional) **show scheduler config**
	Example:
	`switch(config)# show scheduler config`
	`config terminal`
	`feature scheduler`
	`scheduler logfile size 16`
	`end`
		Displays the scheduler configuration.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Defining the Scheduler Log File Size

You can configure the log file size for capturing jobs, schedules, and job output.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**scheduler logfile size** *value* **Example:** `switch(config)# scheduler logfile size 1024`	Defines the scheduler log file size in kilobytes. The range is from 16 to 1024. The default is 16. **Note** If the size of the job output is greater than the size of the log file, then the output is truncated.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Remote User Authentication

You can configure the scheduler to use remote authentication for users who want to configure and schedule jobs.

Note Remote users must authenticate with their clear text password before creating and configuring jobs.

Note Remote user passwords are always shown in encrypted form in the output of the **show running-config** command. The encrypted option (**7**) in the command supports the ASCII device configuration.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**scheduler aaa-authentication password** [**0** | **7**] *password* **Example:** `switch(config)# scheduler aaa-authentication password X12y34Z56a`	Configures a cleartext password for the user who is currently logged in.

	Command or Action	Purpose
Step 3	**scheduler aaa-authentication username** *name* **password** [**0** | **7**] *password* **Example:** `switch(config)# scheduler aaa-authentication username newuser password Z98y76X54b`	Configures a cleartext password for a remote user.
Step 4	(Optional) **show running-config** | **include "scheduler aaa-authentication"** **Example:** `switch(config)# show running-config | include "scheduler aaa-authentication"`	Displays the scheduler password information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Defining a Job

You can define a job including the job name and the command sequence.

Caution After you define a job, you cannot modify or remove commands. To change the job, you must delete it and create a new one.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**scheduler job name** *string* **Example:** `switch(config)# scheduler job name backup-cfg switch(config-job)`	Creates a job and enters the job configuration mode. This example creates a scheduler job named "backup-cfg".
Step 3	*command1* ;[*command2* ;*command3* ;...] **Example:** `switch(config-job)# copy running-config tftp://1.2.3.4/$(SWITCHNAME)-cfg.$(TIMESTAMP)`	Defines the sequence of commands for the specified job. Separate commands with spaces and semicolons (for example, " ;"). This example creates a scheduler job that saves the running configuration to a file in the

	Command or Action	Purpose
	`vrf management` `switch(config-job)#`	bootflash. The job then copies the file from the bootflash to a TFTP server and creates the filename using the current timestamp and switch name.
Step 4	(Optional) **show scheduler job** [**name** *name*] **Example:** `switch(config-job)# show scheduler job`	Displays the job information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting a Job

You can delete a job from the scheduler.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no scheduler job name** *string*
	Example:
	`switch(config)# no scheduler job name configsave`
	`switch(config-job)`
		Deletes the specified job and all commands defined within it.

Step 3	(Optional) **show scheduler job** [**name** *name*] **Example:** `switch(config-job)# show scheduler job name configsave`	Displays the job information.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Defining a Timetable

You can define a timetable in the scheduler to be used with one or more jobs.

If you do not specify the time for the **time** commands, the scheduler assumes the current time. For example, if the current time is March 24, 2013, 22:00 hours, jobs are started as follows:

- For the **time start 23:00 repeat 4:00:00** command, the scheduler assumes a start time of March 24, 2013, 23:00 hours.
- For the **time daily 55** command, the scheduler assumes a start time every day at 22:55 hours.
- For the **time weekly 23:00** command, the scheduler assumes a start time every Friday at 23:00 hours.
- For the **time monthly 23:00** command, the scheduler assumes a start time on the 24th of every month at 23:00 hours.

Note The scheduler will not begin the next occurrence of a job before the last one completes. For example, you have scheduled a job to be completed at one-minute intervals beginning at 22:00; but the job requires two minutes to complete. The scheduler starts the first job at 22:00, completes it at 22:02, and then observes a one-minute interval before starting the next job at 22:03.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**scheduler schedule name** *string*
	Example:
	`switch(config)# scheduler schedule name weekendbackupqos`
	`switch(config-schedule)#`
		Creates a new schedule and places you in schedule configuration mode for that schedule.

Step 3	**job name** *string* **Example:** `switch(config-schedule)# job name offpeakZoning`	Associates a job with this schedule. You can add multiple jobs to a schedule.
Step 4	**time daily** *time* **Example:** `switch(config-schedule)# time daily 23:00`	Indicates the job starts every day at a designated time specified as HH:MM.
Step 5	**time weekly** [[*dow*:]*HH*:]*MM* **Example:** `switch(config-schedule)# time weekly Sun:23:00`	Indicates that the job starts on a specified day of the week. Day of the week (dow) specified as one of the following: • An integer such as 1 = Sunday, 2 = Monday, and so on. • An abbreviation such as Sun = Sunday.

	Command or Action	Purpose
		The maximum length for the entire argument is 10.
Step 6	**time monthly** [[*dm*:]*HH*:]*MM* **Example:** `switch(config-schedule)# time monthly 28:23:00`	Indicates the job starts on a specified day each month (dm). If you specify either 29, 30, or 31, the job is started on the last day of each month.
Step 7	**time start** {**now repeat** *repeat-interval* | *delta-time* [**repeat** *repeat-interval*]} **Example:** `switch(config-schedule)# time start now repeat 48:00`	Indicates the job starts periodically. The start-time format is [[[[yyyy:]mmm:]dd:]HH]:MM. • *delta-time*—Specifies the amount of time to wait after the schedule is configured before starting a job. • **now**—Specifies that the job starts now. • **repeat** *repeat-interval*—Specifies the frequency at which the job is repeated. In this example, the job starts immediately and repeats every 48 hours.
Step 8	(Optional) **show scheduler config** **Example:** `switch(config)# show scheduler config`	Displays the scheduler configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Clearing the Scheduler Log File

You can clear the scheduler log file.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**clear scheduler logfile** **Example:**	Clears the scheduler log file.

	Command or Action	Purpose
	`switch(config)# clear scheduler logfile`	

Verifying the Scheduler Configuration

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

Command	Purpose
show scheduler config	Displays the scheduler configuration.
show scheduler job [**name** *string*]	Displays the jobs configured.
show scheduler logfile	Displays the contents of the scheduler log file.
show scheduler schedule [**name** *string*]	Displays the schedules configured.

Configuration Examples for the Scheduler

Creating a Scheduler Job

This example shows how to create a scheduler job that saves the running configuration to a file in the bootflash. The job then copies the file from the bootflash to a TFTP server (creates the filename using the current timestamp and switch name):

```
switch# configure terminal
switch(config)# scheduler job name backup-cfg
switch(config-job)# copy running-config
tftp://1.2.3.4/$(SWITCHNAME)-cfg.$(TIMESTAMP) vrf management
switch(config-job)# end
switch(config)#
```

Scheduling a Scheduler Job

This example shows how to schedule a scheduler job called backup-cfg to run daily at 1 a.m.:

```
switch# configure terminal
switch(config)# scheduler schedule name daily
switch(config-if)# job name backup-cfg
switch(config-if)# time daily 1:00
switch(config-if)# end
switch(config)#
```

Displaying the Job Schedule

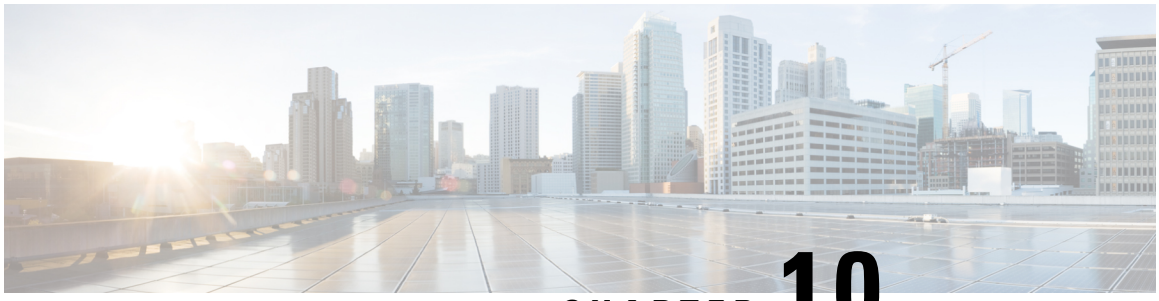
This example shows how to display the job schedule:

```
switch# show scheduler schedule
Schedule Name : daily
---------------------------
User Name : admin
Schedule Type : Run every day at 1 Hrs 00 Mins
Last Execution Time : Fri Jan 2 1:00:00 2013
Last Completion Time: Fri Jan 2 1:00:01 2013
Execution count : 2
-----------------------------------------------
Job Name Last Execution Status
-----------------------------------------------
back-cfg Success (0)
switch#
```

Displaying the Results of Running Scheduler Jobs

This example shows how to display the results of scheduler jobs that have been executed by the scheduler:

```
switch# show scheduler logfile
Job Name : back-cfg Job Status: Failed (1)
Schedule Name : daily User Name : admin
Completion time: Fri Jan 1 1:00:01 2013
-------------------------------- Job Output --------------------------------
`cli var name timestamp 2013-01-01-01.00.00`
`copy running-config bootflash:/$(HOSTNAME)-cfg.$(timestamp)`
`copy bootflash:/switch-cfg.2013-01-01-01.00.00 tftp://1.2.3.4/ vrf management `
copy: cannot access file '/bootflash/switch-cfg.2013-01-01-01.00.00'
==============================================================================
Job Name : back-cfg Job Status: Success (0)
Schedule Name : daily User Name : admin
Completion time: Fri Jan 2 1:00:01 2013
-------------------------------- Job Output --------------------------------
`cli var name timestamp 2013-01-02-01.00.00`
`copy running-config bootflash:/switch-cfg.2013-01-02-01.00.00`
`copy bootflash:/switch-cfg.2013--01-02-01.00.00 tftp://1.2.3.4/ vrf management `
Connection to Server Established.
[ ] 0.50KBTrying to connect to tftp server......
[###### ] 24.50KB
TFTP put operation was successful
==============================================================================
switch#
```



CHAPTER 10

Configuring SNMP

This chapter describes how to configure the SNMP feature on Cisco NX-OS devices.

This chapter contains the following sections:

About SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

SNMP Functional Overview

The SNMP framework consists of three parts:

- An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.
- An SNMP agent—The software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The Cisco Nexus device supports the agent and MIB. To enable the SNMP agent, you must define the relationship between the manager and the agent.
- A managed information base (MIB)—The collection of managed objects on the SNMP agent

SNMP is defined in RFCs 3411 to 3418.

The device supports SNMPv1, SNMPv2c, and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

Cisco NX-OS supports SNMP over IPv6.

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Cisco NX-OS generates SNMP notifications as either traps or informs. A trap is an asynchronous, unacknowledged message sent from the agent to the SNMP managers listed in the host receiver table. Informs are asynchronous messages sent from the SNMP agent to the SNMP manager which the manager must acknowledge receipt of.

Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap. The device cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the device never receives a response, it can send the inform request again.

You can configure Cisco NX-OS to send notifications to multiple host receivers.

The following table lists the SNMP traps that are enabled by default.

Trap Type	Description
generic	: coldStart
entity	: entity_fan_status_change
entity	: entity_mib_change
entity	: entity_module_status_change
entity	: entity_module_inserted
entity	: entity_module_removed
entity	: entity_power_out_change
entity	: entity_power_status_change
entity	: entity_unrecognised_module
link	: cErrDisableInterfaceEventRev1
link	: cieLinkDown
link	: cieLinkUp
link	: cmn-mac-move-notification
link	: delayed-link-state-change
link	: extended-linkDown
link	: extended-linkUp
link	: linkDown
link	: linkUp
rf	: redundancy_framework
license	: notify-license-expiry

Trap Type	Description
license	: notify-no-license-for-feature
license	: notify-licensefile-missing
license	: notify-license-expiry-warning
upgrade	: UpgradeOpNotifyOnCompletion
upgrade	: UpgradeJobStatusNotify
entity	: entity_sensor
rmon	: fallingAlarm
rmon	: hcRisingAlarm
rmon	: hcFallingAlarm
rmon	: risingAlarm

SNMPv3

SNMPv3 provides secure access to devices by a combination of authenticating and encrypting frames over the network. The security features provided in SNMPv3 are the following:

- Message integrity—Ensures that a packet has not been tampered with in-transit.
- Authentication—Determines the message is from a valid source.
- Encryption—Scrambles the packet contents to prevent it from being seen by unauthorized sources.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

Security Models and Levels for SNMPv1, v2, v3

The security level determines if an SNMP message needs to be protected from disclosure and if the message needs to be authenticated. The various security levels that exist within a security model are as follows:

- noAuthNoPriv—Security level that does not provide authentication or encryption. This level is not supported for SNMPv3.
- authNoPriv—Security level that provides authentication but does not provide encryption.
- authPriv—Security level that provides both authentication and encryption.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The security model combined with the security level determine the security mechanism applied when the SNMP message is processed. The following table identifies what the combinations of security models and levels mean.

Table 9: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	authNoPriv	HMAC-MD5, or HMAC-SHA	No	Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA).
v3	authPriv	HMAC-MD5, or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard.

User-Based Security Model

The SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur nonmaliciously.
- Message origin authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- Message confidentiality—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages.

Cisco NX-OS uses three authentication protocols for SNMPv3:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol

Cisco NX-OS uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The **priv** option offers a choice of DES or 128-bit AES encryption for SNMP security encryption. The **priv** option and the **aes-128** token indicate that this privacy password is for generating a 128-bit AES key. The AES priv password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 case-sensitive, alphanumeric characters. If you use the localized key, you can specify a maximum of 130 characters.

Note For an SNMPv3 operation using the external AAA server, you must use AES for the privacy protocol in the user configuration on the external AAA server.

CLI and SNMP User Synchronization

SNMPv3 user management can be centralized at the Access Authentication and Accounting (AAA) server level. This centralized user management allows the SNMP agent in Cisco NX-OS to leverage the user authentication service of the AAA server. Once user authentication is verified, the SNMP PDUs are processed further. Additionally, the AAA server is also used to store user group names. SNMP uses the group names to apply the access/role policy that is locally available in the switch.

Any configuration changes made to the user group, role, or password results in database synchronization for both SNMP and AAA.

Cisco NX-OS synchronizes the user configuration in the following ways:

- The authentication passphrase specified in the **snmp-server user** command becomes the password for the CLI user.
- The password specified in the **username** command becomes the authentication and privacy passphrases for the SNMP user.
- If you create or delete a user using either SNMP or the CLI, the user is created or deleted for both SNMP and the CLI.
- User-role mapping changes are synchronized in SNMP and the CLI.
- Role changes (deletions or modifications) from the CLI are synchronized to SNMP.

Note When you configure a passphrase/password in localized key/encrypted format, Cisco NX-OS does not synchronize the user information (passwords, roles, and so on).

Cisco NX-OS holds the synchronized user configuration for 60 minutes by default.

Group-Based SNMP Access

Note Because *group* is a standard SNMP term used industry-wide, we refer to roles as groups in this SNMP section.

SNMP access rights are organized by groups. Each group in SNMP is similar to a role through the CLI. Each group is defined with read access or read-write access.

You can begin communicating with the agent once your username is created, your roles are set up by your administrator, and you are added to the roles.

SNMP and Embedded Event Manager

The Embedded Event Manager (EEM) feature monitors events, including SNMP MIB objects, and triggers an action based on these events. One of the actions could be to send an SNMP notification. EEM sends the cEventMgrPolicyEvent of CISCO-EMBEDDED-EVENT-MGR-MIB as the SNMP notification.

Multiple Instance Support

A device can support multiple instances of a logical network entity, such as protocol instances or virtual routing and forwarding (VRF) instances. Most existing MIBs cannot distinguish between these multiple logical network entities. For example, the original OSPF-MIB assumes a single protocol instance on a device, but you can now configure multiple OSPF instances on a device.

SNMPv3 uses contexts to distinguish between these multiple instances. An SNMP context is a collection of management information that you can access through the SNMP agent. A device can support multiple contexts for different logical network entities. An SNMP context allows the SNMP manager to access one of the multiple instances of a MIB module supported on the device for the different logical network entities.

Cisco NX-OS supports the CISCO-CONTEXT-MAPPING-MIB to map between SNMP contexts and logical network entities. You can associate an SNMP context to a VRF, protocol instance, or topology.

SNMPv3 supports contexts with the contextName field of the SNMPv3 PDU. You can map this contextName field to a particular protocol instance or VRF.

For SNMPv2c, you can map the SNMP community to a context using the snmpCommunityContextName MIB object in the SNMP-COMMUNITY-MIB (RFC 3584). You can then map this snmpCommunityContextName to a particular protocol instance or VRF using the CISCO-CONTEXT-MAPPING-MIB or the CLI.

High Availability for SNMP

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

Virtualization Support for SNMP

Cisco NX-OS supports one instance of the SNMP. SNMP supports multiple MIB module instances and maps them to logical network entities.

SNMP is also VRF aware. You can configure SNMP to use a particular VRF to reach the SNMP notification host receiver. You can also configure SNMP to filter notifications to an SNMP host receiver based on the VRF where the notification occurred.

Guidelines and Limitations for SNMP

SNMP has the following configuration guidelines and limitations:

- Access control list (ACLs) can be applied only to local SNMPv3 users configured on the switch. ACLs cannot be applied to remote SNMPv3 users stored on Authentication, Authorization, and Accounting (AAA) servers.
- Cisco NX-OS supports read-only access to some SNMP MIBs. See the Cisco NX-OS MIB support list at the following URL for more information: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html
- Cisco NX-OS does not support the SNMPv3 noAuthNoPriv security level.
- Special characters @ and % are not allowed in the SNMP community string.
- The default SNMP PDU value is 1500 bytes. The SNMP agent drops any response PDU that is greater than 1500 bytes, causing the SNMP request to fail. To receive MIB data values larger than 1500 bytes, use the **snmp-server packetsize** <*byte-count*> command to reconfigure the packet size. The valid byte-count range is from 484 to 17382. When a GETBULK response exceeds the packet size, the data can get truncated.
- You must use either the CLI or SNMP to configure a feature on your switch. Do not configure a feature using both interfaces to the switch.
- Using cefcFanTrayOperStatus snmpwalk on an individual fan OID tree where the fan is not populated in chassis, can return a response for next OID entry in the tree. To prevent this behavior, use the -CI option in snmpwalk.

  The behavior is not seen when polling parent OID, or when using getmany.

Default Settings for SNMP

The following table lists the default settings for SNMP parameters.

Parameters	Default
License notifications	Enabled

Configuring SNMP

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Configuring SNMP Users

You can configure a user for SNMP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show snmp user** **Example:** `switch(config)# show snmp user`	Displays information about one or more SNMP users.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enforcing SNMP Message Encryption

You can configure SNMP to require authentication or encryption for incoming requests. By default, the SNMP agent accepts SNMPv3 messages without authentication and encryption. When you enforce privacy, Cisco NX-OS responds with an authorization error for any SNMPv3 PDU request using a security level parameter of either **noAuthNoPriv** or **authNoPriv**.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server user** *name* **enforcePriv** **Example:** `switch(config)# snmp-server user Admin enforcePriv`	Enforces SNMP message encryption for this user.
Step 3	**snmp-server globalEnforcePriv** **Example:** `switch(config)# snmp-server globalEnforcePriv`	Enforces SNMP message encryption for all users.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Assigning SNMPv3 Users to Multiple Roles

After you configure an SNMP user, you can assign multiple roles for the user.

Note Only users belonging to a network-admin role can assign roles to other users.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server user** *name group* **Example:** `switch(config)# snmp-server user Admin superuser`	Associates this SNMP user with the configured user role.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**snmp-server community** *name* {**group** *group* | **ro** | **rw**} **Example:** `switch(config)# snmp-server community public ro`	Creates an SNMP community string.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Filtering SNMP Requests

ACL with SNMPv3 user is not supported. You can assign an access control list (ACL) to an SNMPv2 community to filter SNMP requests. If the assigned ACL allows the incoming request packet, SNMP processes the request. If the ACL denies the request, SNMP drops the request and sends a system message.

Create the ACL with the following parameters:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Protocol (UDP or TCP)

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server community** *name* [**use-ipv4acl** *acl-name*] **Example:** `switch(config)# snmp-server community public use-ipv4acl myacl`	Assigns an IPv4 ACL to an SNMPv2 community to filter SNMP requests.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SNMP Notification Receivers

You can configure Cisco NX-OS to generate SNMP notifications to multiple host receivers.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server host** *ip-address* **traps version 1** *community* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 traps version 1 public`	Configures a host receiver for SNMPv1 traps. The *ip-address* can be an IPv4 or IPv6 address. The *community* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.
Step 3	**snmp-server host** *ip-address* {**traps** | **informs**} **version 2c** *community* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 informs version 2c public`	Configures a host receiver for SNMPv2c traps or informs. The *ip-address* can be an IPv4 or IPv6 address. The *community* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.
Step 4	**snmp-server host** *ip-address* {**traps** | **informs**} **version 3** {**auth** | **noauth** | **priv**} *username* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS`	Configures a host receiver for SNMPv3 traps or informs. The *ip-address* can be an IPv4 or IPv6 address. The *username* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. **Note** The SNMP manager must know the user credentials (authKey/PrivKey) based on the SNMP engine ID of the Cisco NX-OS device to authenticate and decrypt the SNMPv3 messages.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Source Interface for SNMP Notifications

You can configure SNMP to use the IP address of an interface as the source IP address for notifications. When a notification is generated, its source IP address is based on the IP address of this configured interface.

You can configure a source interface as follows:

- All notifications sent to all SNMP notification receivers.
- All notifications sent to a specific SNMP notification receiver. This configuration overrides the global source interface configuration.

Note Configuring the source interface IP address for outgoing trap packets does not guarantee that the device will use the same interface to send the trap. The source interface IP address defines the source address inside of the SNMP trap, and the connection is opened with the address of the egress interface as source.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server host** *ip-address* **source-interface** *if-type if-number* **traps version 2c** *name* **Example:** `snmp-server host 192.0.2.1 source-interface ethernet 2/1 traps version 2c public`	(Optional) Send Traps messages to this host. The traps version is the SNMP version to use for notification messages. 2c indicates that SNMPv2c is to be used.
Step 3	**snmp-server host** *ip-address* **source-interface** *if-type if-number* **use-vrf** *vrf-name* **Example:** `snmp-server host 192.0.2.1 source-interface ethernet 2/1 use-vrf default`	Configures SNMP to use the selected VRF to communicate with the host receiver. The ip-address can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 32 characters. **Note** This command does not remove the host configuration.
Step 4	**snmp-server host** *ip-address* **source-interface** *if-type if-number* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 source-interface ethernet 2/1`	Configures a host receiver for SNMPv2c traps or informs. The *ip-address* can be an IPv4 or IPv6 address. Use **?** to determine the supported interface types. The UDP port number range is from 0 to 65535. This configuration overrides the global source interface configuration.
Step 5	**snmp-server source-interface** {**traps** | **informs**} *if-type if-number* **Example:** `switch(config)# snmp-server source-interface traps ethernet 2/1`	Configures a source interface for sending out SNMPv2c traps or informs. Use **?** to determine the supported interface types.

	Command or Action	Purpose
Step 6	**show snmp source-interface** **Example:** `switch(config)# show snmp source-interface`	Displays information about configured source interfaces.

Configuring the Notification Target User

You must configure a notification target user on the device to send SNMPv3 inform notifications to a notification host receiver.

Cisco NX-OS uses the credentials of the notification target user to encrypt the SNMPv3 inform notification messages to the configured notification host receiver.

Note For authenticating and decrypting the received inform PDU, the notification host receiver should have the same user credentials as configured in Cisco NX-OS to authenticate and decrypt the informs.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server user** *name* [**auth** {**md5** | **sha** | **sha-256**} *passphrase* [**auto**] [**priv [aes-128]** *passphrase*] [**engineID** *id*] **Example:** `switch(config)# snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID 00:00:00:63:00:01:00:10:20:15:10:03`	Configures the notification target user with the specified engine ID for the notification host receiver. The engine ID format is a 12-digit colon-separated decimal number.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SNMP Notification Receivers with VRFs

SNMP adds entries into the cExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MIB when you configure the VRF reachability and filtering options for an SNMP notification receiver.

Note You must configure the host before configuring the VRF reachability or filtering options.

You can configure Cisco NX-OS to use a configured VRF to reach the host receiver or to filter notifications based on the VRF in which the notification occurred.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 use-vrf Blue`	Configures SNMP to use the selected VRF to communicate with the host receiver. The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. The **no** form of this command removes the VRF reachability information for the configured host and removes the entry from the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. **Note** This command does not remove the host configuration.
Step 3	[**no**] **snmp-server host** *ip-address* **filter-vrf** *vrf-name* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 filter-vrf Red`	Filters notifications to the notification host receiver based on the configured VRF. The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. The **no** form of this command removes the VRF filter information for the configured host and removes the entry from the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. **Note** This command does not remove the host configuration.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SNMP to Send Traps Using an Inband Port

You can configure SNMP to send traps using an inband port. To do so, you must configure the source interface (at the global or host level) and the VRF used to send the traps.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server source-interface traps** *if-type if-number* **Example:** `switch(config)# snmp-server source-interface traps ethernet 1/2`	Globally configures a source interface for sending out SNMP traps. Use ? to determine the supported interface types. You can configure the source interface at the global level or at a host level. When the source interface is configured globally, any new host configuration uses the global configuration to send the traps. **Note** To configure a source interface at the host level, use the **snmp-server host** *ip-address* **source-interface** *if-type if-number* command.
Step 3	(Optional) **show snmp source-interface** **Example:** `switch(config)# show snmp source-interface`	Displays information about configured source interfaces.
Step 4	**snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 171.71.48.164 use-vrf default`	Configures SNMP to use the selected VRF to communicate with the host receiver. The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

	Command or Action	Purpose
		Note By default, SNMP sends the traps using the management VRF. If you do not want to use the management VRF, you must use this command to specify the desired VRF.
Step 5	(Optional) **show snmp host** **Example:** `switch(config)# show snmp host`	Displays information about configured SNMP hosts.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling SNMP Notifications

You can enable or disable notifications. If you do not specify a notification name, Cisco NX-OS enables all notifications except BGP, EIGRP, and OSPF notifications.

Note The **snmp-server enable traps** command enables both traps and informs, depending on the configured notification host receivers.

The following table lists the commands that enable the notifications for Cisco NX-OS MIBs.

Table 10: Enabling SNMP Notifications

MIB	Related Commands
All notifications (except BGP, EIGRP, and OSPF)	**snmp-server enable traps**
CISCO-AAA-SERVER-MIB	**snmp-server enable traps aaa** **snmp-server enable traps aaa server-state-change**
CISCO-BGP4-MIB	**snmp-server enable traps bgp**
CISCO-CALLHOME-MIB	**snmp-server enable traps callhome** **snmp-server enable traps callhome event-notify** **snmp-server enable traps callhome smtp-send-fail**
CISCO-CONFIG-MAN-MIB	**snmp-server enable traps config** **snmp-server enable traps config ccmCLIRunningConfigChanged**

MIB	Related Commands
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp** [*tag*]
CISCO-ERR-DISABLE-MIB	**snmp-server enable traps link cerrDisableInterfaceEventRev1**
ENTITY-MIB, CISCO-ENTITY-SENSOR-MIB	**snmp-server enable traps entity** **snmp-server enable traps entity entity_fan_status_change** **snmp-server enable traps entity entity_mib_change** **snmp-server enable traps entity entity_module_inserted** **snmp-server enable traps entity entity_module_removed** **snmp-server enable traps entity entity_module_status_change** **snmp-server enable traps entity entity_power_out_change** **snmp-server enable traps entity entity_power_status_change** **snmp-server enable traps entity entity_unrecognised_module**
CISCO-FEATURE-CONTROL-MIB	**snmp-server enable traps feature-control** **snmp-server enable traps feature-control FeatureOpStatusChange**
CISCO-HSRP-MIB	**snmp-server enable traps hsrp** **snmp-server enable traps hsrp state-change**
CISCO-LICENSE-MGR-MIB	**snmp-server enable traps license** **snmp-server enable traps license notify-license-expiry** **snmp-server enable traps license notify-license-expiry-warning** **snmp-server enable traps license notify-licensefile-missing** **snmp-server enable traps license notify-no-license-for-feature**

MIB	Related Commands
IF-MIB	**snmp-server enable traps link** **snmp-server enable traps link IETF-extended-linkDown** **snmp-server enable traps link IETF-extended-linkUp** **snmp-server enable traps link cisco-extended-linkDown** **snmp-server enable traps link cisco-extended-linkUp** **snmp-server enable traps link linkDown** **snmp-server enable traps link Up**
OSPF-MIB, OSPF-TRAP-MIB	**snmp-server enable traps ospf** [*tag*] **snmp-server enable traps ospf lsa** **snmp-server enable traps ospf rate-limit** *rate*
CISCO-RF-MIB	**snmp-server enable traps rf** **snmp-server enable traps rf redundancy_framework**
CISCO-RMON-MIB	**snmp-server enable traps rmon** **snmp-server enable traps rmon fallingAlarm** **snmp-server enable traps rmon hcFallingAlarm** **snmp-server enable traps rmon hcRisingAlarm** **snmp-server enable traps rmon risingAlarm**
SNMPv2-MIB	**snmp-server enable traps snmp** **snmp-server enable traps snmp authentication**
CISCO-PORT-STORM-CONTROL-MIB	**storm-control action trap**
CISCO-STP-EXTENSIONS-MIB	**snmp-server enable traps stpx stpxMstInconsistencyUpdate**
CISCO-STP-BRIDGE-MIB	**snmp-server enable traps bridge** **snmp-server enable traps bridge newroot** **snmp-server enable traps bridge topologychange**

MIB	Related Commands
CISCO-STPX-MIB	**snmp-server enable traps stpx** **snmp-server enable traps stpx inconsistency** **snmp-server enable traps stpx loop-inconsistency** **snmp-server enable traps stpx root-inconsistency**
CISCO-SYSTEM-EXT-MIB	**snmp-server enable traps sysmgr** **snmp-server enable traps sysmgr cseFailSwCoreNotifyExtended**
UPGRADE-MIB	**snmp-server enable traps upgrade** **snmp-server enable traps upgrade UpgradeJobStatusNotify** **snmp-server enable traps upgrade UpgradeOpNotifyOnCompletion**
VTP-MIB	**snmp-server enable traps vtp** **snmp-server enable traps vtp notifs** **snmp-server enable traps vtp vlancreate** **snmp-server enable traps vtp vlandelete**

Use the following commands in the configuration mode shown to enable the specified notification:

Command	Purpose
snmp-server enable traps **Example:** `switch(config)# snmp-server enable traps`	Enables all SNMP notifications.
snmp-server enable traps aaa [**server-state-change**] **Example:** `switch(config)# snmp-server enable traps aaa`	Enables the AAA SNMP notifications. Optionally, enables the following specific notifications: • **server-state-change**—Enables AAA server state-change notifications.
snmp-server enable traps bgp **Example:** `switch(config)# snmp-server enable traps bgp`	Enables Border Gateway Protocol (BGP) SNMP notifications.
snmp-server enable traps bridge [**newroot**] [**topologychange**] **Example:** `switch(config)# snmp-server enable traps bridge`	Enables STP bridge SNMP notifications. Optionally, enables the following specific notifications: • **newroot**—Enables STP new root bridge notifications. • **topologychange**—Enables STP bridge topology-change notifications.

Command	Purpose
snmp-server enable traps callhome [**event-notify**] [**smtp-send-fail**] **Example:** `switch(config)# snmp-server enable traps callhome`	Enables Call Home notifications. Optionally, enables the following specific notifications: • **event-notify**—Enables Call Home external event notifications. • **smtp-send-fail**—Enables Simple Mail Transfer Protocol (SMTP) message send fail notifications.
snmp-server enable traps config [**ccmCLIRunningConfigChanged**] **Example:** `switch(config)# snmp-server enable traps config`	Enables SNMP notifications for configuration changes. • **ccmCLIRunningConfigChanged**—Enables SNMP notifications for configuration changes in the running or startup configuration.
snmp-server enable traps eigrp [*tag*] **Example:** `switch(config)# snmp-server enable traps eigrp`	Enables CISCO-EIGRP-MIB SNMP notifications.
snmp-server enable traps entity [**entity_fan_status_change**] [**entity_mib_change**] [**entity_module_inserted**] [**entity_module_removed**] [**entity_module_status_change**] [**entity_power_out_change**] [**entity_power_status_change**] [**entity_unrecognised_module**] **Example:** `switch(config)# snmp-server enable traps entity`	Enables ENTITY-MIB SNMP notifications. Optionally, enables the following specific notifications: • **entity_fan_status_change**—Enables entity fan status-change notifications. • **entity_mib_change**—Enables entity MIB change notifications. • **entity_module_inserted**—Enables entity module inserted notifications. • **entity_module_removed**—Enables entity module removed notifications. • **entity_module_status_change**—Enables entity module status-change notifications. • **entity_power_out_change**—Enables entity power-out change notifications. • **entity_power_status_change**—Enables entity power status-change notifications. • **entity_unrecognised_module**—Enables entity unrecognized module notifications.

Command	Purpose
snmp-server enable traps feature-control [**FeatureOpStatusChange**] **Example:** `switch(config)# snmp-server enable traps feature-control`	Enables feature-control SNMP notifications. Optionally, enables the following specific notifications: • **FeatureOpStatusChange**—Enables feature operation status-change notifications.
snmp-server enable traps hsrp state-change **Example:** `switch(config)# snmp-server enable traps hsrp`	Enables CISCO-HSRP-MIB SNMP notifications. Optionally, enables the following specific notifications: • **state-change**—Enables HSRP state-change notifications.
snmp-server enable traps license [**notify-license-expiry**] [**notify-license-expiry-warning**] [**notify-licensefile-missing**] [**notify-no-license-for-feature**] **Example:** `switch(config)# snmp-server enable traps license`	Enables ENTITY-MIB SNMP notifications. Optionally, enables the following specific notifications: • **notify-license-expiry**—Enables license expiry notifications. • **notify-license-expiry-warning**—Enables license expiry warning notifications. • **notify-licensefile-missing**—Enables license file-missing notifications. • **notify-no-license-for-feature**—Enables no-license-installed-for-feature notifications.

Command	Purpose
snmp-server enable traps link [**cieLinkDown**] [**cieLinkUp**] [**cmn-mac-move-notification**] [**IETF-extended-linkDown**] [**IETF-extended-linkUp**] [**cisco-extended-linkDown**] [**cisco-extended-linkUp**][**linkDown**] [**linkUp**] **Example:** `switch(config)# snmp-server enable traps link`	Enables IF-MIB link notifications. Optionally, enable the following specific notifications: • **IETF-extended-linkDown**—Enables Cisco extended link state down notifications. • **IETF-extended-linkUp**—Enables Cisco extended link state up notifications. • **cmn-mac-move-notification**—Enables MAC address move notifications. • **cisco-extended-linkDown**—Enables Internet Engineering Task Force (IETF) extended link state down notifications. • **cisco-extended-linkUp**—Enables Internet Engineering Task Force (IETF) extended link state up notifications. • **linkDown**—Enables IETF link state down notifications. • **linkUp**—Enables IETF link state up notifications.
snmp-server enable traps ospf [*tag*] [**lsa**] **Example:** `switch(config)# snmp-server enable traps ospf`	Enables Open Shortest Path First (OSPF) notifications. Optionally, enables the following specific notifications: • **lsa**—Enables OSPF link state advertisement (LSA) notifications.
snmp-server enable traps rf [**redundancy-framework**] **Example:** `switch(config)# snmp-server enable traps rf`	Enables redundancy framework (RF) SNMP notifications. Optionally, enables the following specific notifications: • **redundancy-framework**—Enables RF supervisor switchover MIB notifications.

Command	Purpose
snmp-server enable traps rmon [**fallingAlarm**] [**hcFallingAlarm**] [**hcRisingAlarm**] [**risingAlarm**] **Example:** `switch(config)# snmp-server enable traps rmon`	Enables remote monitoring (RMON) SNMP notifications. Optionally, enables the following specific notifications: • **fallingAlarm**—Enables RMON falling alarm notifications. • **hcFallingAlarm**—Enables RMON high-capacity falling alarm notifications. • **hcRisingAlarm**—Enables RMON high-capacity rising alarm notifications. • **risingAlarm**—Enables RMON rising alarm notifications.
snmp-server enable traps snmp [**authentication**] **Example:** `switch(config)# snmp-server enable traps snmp`	Enables general SNMP notifications. Optionally, enables the following specific notifications: • **authentication**—Enables SNMP authentication notifications.
snmp-server enable traps stpx [**inconsistency**] [**loop-inconsistency**] [**root-inconsistency**] **Example:** `switch(config)# snmp-server enable traps stpx`	Enables SNMP STPX notifications. Optionally, enables the following specific notifications: • **inconsistency**—Enables SNMP STPX MIB inconsistency update notifications. • **loop-inconsistency**—Enables SNMP STPX MIB loop-inconsistency update notifications. • **root-inconsistency**—Enables SNMP STPX MIB root-inconsistency update notifications.
snmp-server enable traps syslog [**message-generated**] **Example:** `switch(config)# snmp-server enable traps syslog`	Sends syslog messages as traps to the defined SNMP host. Optionally, enables the following specific notifications: • **message-generated**—Enables software log message generated notifications.
snmp-server enable traps sysmgr [**cseFailSwCoreNotifyExtended**] **Example:** `switch(config)# snmp-server enable traps sysmgr`	Enables software change notifications. Optionally, enables the following specific notifications: • **cseFailSwCoreNotifyExtended**—Enables software core notifications.

Command	Purpose
snmp-server enable traps upgrade [**UpgradeJobStatusNotify**] [**UpgradeOpNotifyOnCompletion**] **Example:** `switch(config)# snmp-server enable traps upgrade`	Enables upgrade notifications. Optionally, enables the following specific notifications: • **UpgradeJobStatusNotify**—Enables upgrade job status notifications. • **UpgradeOpNotifyOnCompletion**—Enables upgrade global status notifications.
snmp-server enable traps vtp [**notifs**] [**vlancreate**] [**vlandelete**] **Example:** `switch(config)# snmp-server enable traps vtp`	Enables VTP notifications. Optionally, enables the following specific notifications: • **notifs**—Enables VTP notifications. • **vlancreate**—Enables VLAN creation notifications. • **vlandelete**—Enables VLAN deletion notifications.
storm-control action traps **Example:** `switch(config-if)# storm-control action traps`	Enables traffic storm control notifications when the traffic storm control limit is reached.

Disabling Link Notifications on an Interface

You can disable linkUp and linkDown notifications on an individual interface. You can use this limit notifications on flapping interface (an interface that transitions between up and down repeatedly).

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type slot*/*port* **Example:** `switch(config)# interface ethernet 2/2`	Disables SNMP link-state traps for the interface. This command is enabled by default.
Step 3	**no snmp trap link-status** **Example:** `switch(config-if)# no snmp trap link-status`	Disables SNMP link-state traps for the interface. This command is enabled by default.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Displaying SNMP ifIndex for an Interface

The SNMP ifIndex is used across multiple SNMP MIBs to link related interface information.

Procedure

	Command or Action	Purpose

Step 1	**show interface snmp-ifindex**
	Example:
	`switch# show interface snmp-ifindex | grep -i Eth12/1`
	`Eth12/1 441974784 (0x1a580000)`
		Displays the persistent SNMP ifIndex value from the IF-MIB for all interfaces. Optionally, use the **|** keyword and the **grep** keyword to search for a particular interface in the output.

Enabling a One-Time Authentication for SNMP over TCP

You can enable a one-time authentication for SNMP over a TCP session.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server tcp-session** [**auth**] **Example:** `switch(config)# snmp-server tcp-session`	Enables a one-time authentication for SNMP over a TCP session. The default is disabled.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Assigning SNMP Device Contact and Location Information

You can assign the device contact information, which is limited to 32 characters (without spaces) and the device location.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server contact** *name* **Example:** `switch(config)# snmp-server contact Admin`	Configures sysContact, which is the SNMP contact name.
Step 3	**snmp-server location** *name* **Example:** `switch(config)# snmp-server location Lab-7`	Configures sysLocation, which is the SNMP location.
Step 4	(Optional) **show snmp** **Example:** `switch(config)# show snmp`	Displays information about one or more destination profiles.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Context to Network Entity Mapping

You can configure an SNMP context to map to a logical network entity, such as a protocol instance or VRF.

Before you begin

Determine the logical network entity instance. For more information on VRFs and protocol instances, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide or the Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]	Maps an SNMP context to a protocol instance, VRF, or topology. The names can be any alphanumeric string up to 32 characters.

	Command or Action	Purpose
	Example: `switch(config)# snmp-server context public1 vrf red`	The **no** option deletes the mapping between an SNMP context and a protocol instance, VRF, or topology. **Note** Do not enter an instance, VRF, or topology to delete a context mapping. If you use the instance, VRF, or topology keywords, you configure a mapping between the context and a zero-length string.
Step 3	(Optional) **snmp-server mib community-map** *community-name* **context** *context-name* **Example:** `switch(config)# snmp-server mib community-map public context public1`	Maps an SNMPv2c community to an SNMP context. The names can be any alphanumeric string up to 32 characters.
Step 4	(Optional) **show snmp context** **Example:** `switch(config)# show snmp context`	Displays information about one or more SNMP contexts.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling SNMP

You can disable SNMP on the device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no snmp-server protocol enable** **Example:** `switch(config)# no snmp-server protocol enable`	Disables SNMP. SNMP is enabled by default. **Note** You cannot disable SNMPv1 without disabling SNMPv2. If you want to disable SNMPv1, then configure only SNMPv3, or disable SNMP entirely.

Modifying the AAA Synchronization Time

You can modify how long Cisco NX-OS holds the synchronized user configuration.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**snmp-server aaa-user cache-timeout** *seconds* **Example:** `switch(config)# snmp-server aaa-user cache-timeout 1200`	Configures how long the AAA synchronized user configuration stays in the local cache. The range is from 1 to 86400 seconds. The default is 3600.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying SNMP Configuration

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

Command	Purpose
show interface snmp-ifindex	Displays the SNMP ifIndex value for all interfaces (from IF-MIB).
show running-config snmp [**all**]	Displays the SNMP running configuration. Beginning with 9.3(8) release, SNMPv3 users under show run will be represented in SALT format instead of hash.
show snmp	Displays the SNMP status.

Command	Purpose
show snmp community	Displays the SNMP community strings. **Note** If the name of the SNMP context in the **snmp-server mib community-map** command is more than 11 characters, the output of the **show snmp community** command is displayed in a vertical format instead of a tabular format.
show snmp context	Displays the SNMP context mapping.
show snmp engineID	Displays the SNMP engineID.
show snmp group	Displays SNMP roles.
show snmp host	Displays information about configured SNMP hosts.
show snmp session	Displays SNMP sessions.
show snmp source-interface	Displays information about configured source interfaces.
show snmp trap	Displays the SNMP notifications enabled or disabled.
show snmp user	Displays SNMPv3 users.

Configuration Examples for SNMP

This example shows how to configure Cisco NX-OS to send the Cisco linkUp or Down notifications to one notification host receiver using the Blue VRF and defines two SNMP users, Admin and NMS:

```
configure terminal
snmp-server contact Admin@company.com
snmp-server user Admin auth sha abcd1234 priv abcdefgh
snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID
00:00:00:63:00:01:00:22:32:15:10:03
snmp-server host 192.0.2.1 informs version 3 auth NMS
snmp-server host 192.0.2.1 use-vrf Blue
snmp-server enable traps link cisco
```

This example shows how to configure SNMP to send traps using an inband port configured at the host level:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server host 171.71.48.164 version 2c public
switch(config)# snmp-server host 171.71.48.164 source-interface ethernet 1/2
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Source interface: Ethernet 1/2
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use-vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-------------------------------------------------------------------
```

This example shows how to configure SNMP to send traps using a globally configured inband port:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server source-interface traps ethernet 1/2
switch(config)# show snmp source-interface
-------------------------------------------------------------------
Notification source-interface
-------------------------------------------------------------------
trap Ethernet1/2
inform -
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use_vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-------------------------------------------------------------------
```

This example shows how to map VRF red to the SNMPv2c public community string:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vrf context red
switch(config-vrf)# exit
switch(config)# snmp-server context public1 vrf red
switch(config)# snmp-server mib community-map public context public1
```

This example shows how to map OSPF instance Enterprise to the same SNMPv2c public community string:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature ospf
switch(config)# router ospf Enterprise
switch(config-router)# exit
switch(config)# snmp-server context public1 instance Enterprise
```

```
switch(config)# snmp-server mib community-map public context public1
```

Additional References

Related Documents

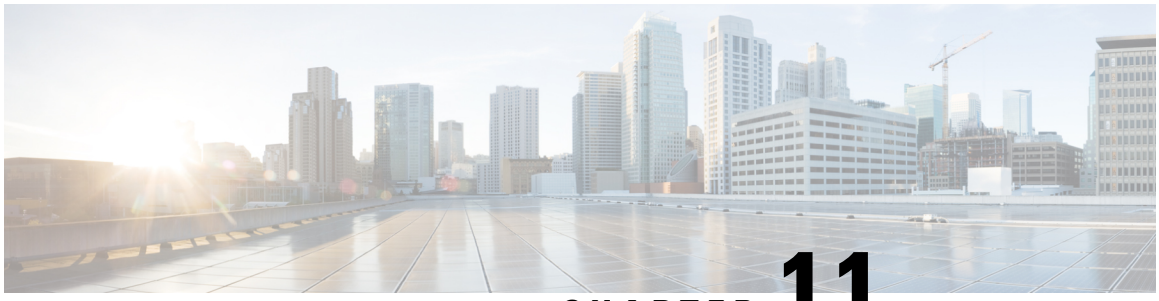
Related Topic	Document Title
IP ACLs and AAA	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
MIBs	*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*

RFCs

RFC	Title
RFC 3414	*User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

MIBs

MIBs	MIBs Link
MIBs related to SNMP	To locate and download supported MIBs, go to the follow ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 11

Configuring RMON

This chapter describes how to configure the remote monitoring (RMON) feature on Cisco NX-OS devices.

This chapter contains the following sections:

About RMON

RMON is a Simple Network Management Protocol (SNMP) Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. Cisco NX-OS supports RMON alarms, events, and logs to monitor Cisco NX-OS devices.

An RMON alarm monitors a specific management information base (MIB) object for a specified interval, triggers an alarm at a specified threshold value (threshold), and resets the alarm at another threshold value. You can use alarms with RMON events to generate a log entry or an SNMP notification when the RMON alarm triggers.

RMON is enabled by default, but no alarms are configured in Cisco NX-OS. You can configure RMON alarms by using the CLI or an SNMP-compatible network management station.

RMON Alarms

You can set an alarm on any MIB object that resolves into an SNMP INTEGER type. The specified object must be an existing SNMP MIB object in standard dot notation (for example, 1.3.6.1.2.1.2.2.1.14 represents ifInOctets.14).

When you create an alarm, you specify the following parameters:

- MIB object to monitor.
- Sampling interval—The interval that the device uses to collect a sample value of the MIB object.

- Sample type—Absolute samples take the current snapshot of the MIB object value. Delta samples take two consecutive samples and calculate the difference between them.
- Rising threshold—The value at which the device triggers a rising alarm or resets a falling alarm.
- Falling threshold—The value at which the device triggers a falling alarm or resets a rising alarm.
- Events—The action that the device takes when an alarm (rising or falling) triggers.

Note Use the hcalarms option to set an alarm on a 64-bit integer MIB object.

For example, you can set a delta type rising alarm on an error counter MIB object. If the error counter delta exceeds this value, you can trigger an event that sends an SNMP notification and logs the rising alarm event. This rising alarm will not occur again until the delta sample for the error counter drops below the falling threshold.

Note The falling threshold must be less than the rising threshold.

RMON Events

You can associate a particular event to each RMON alarm. RMON supports the following event types:

- SNMP notification—Sends an SNMP risingAlarm or fallingAlarm notification when the associated alarm triggers.
- Log—Adds an entry in the RMON log table when the associated alarm triggers.
- Both—Sends an SNMP notification and adds an entry in the RMON log table when the associated alarm triggers.

You can specify a different even for a falling alarm and a rising alarm.

Note You may choose to use the default RMON events template configuration or you can delete these entries and create new RMON events. Until you create RMON alarm configurations, no alarms will be triggered by these configurations.

High Availability for RMON

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

Virtualization Support for RMON

Cisco NX-OS supports one instance of RMON.

RMON is virtual routing and forwarding (VRF) aware. You can configure RMON to use a particular VRF to reach the RMON SMTP server.

Guidelines and Limitations for RMON

RMON has the following configuration guidelines and limitations:

- You must configure an SNMP user and a notification receiver to use the SNMP notification event type.
- You can configure an RMON alarm only on a MIB object that resolves to an integer.
- When you configure an RMON alarm, the object identifier must be complete with its index so that it refers to only one object. For example, 1.3.6.1.2.1.2.2.1.14 corresponds to cpmCPUTotal5minRev, and .1 corresponds to index cpmCPUTotalIndex, which creates object identifier 1.3.6.1.2.1.2.2.1.14.1.

Default Settings for RMON

The following table lists the default settings for RMON parameters.

Parameters	Default
RMON	Enabled
Alarms	None configured

Configuring RMON

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Configuring RMON Alarms

You can configure RMON alarms on any integer-based SNMP MIB object.

You can optionally specify the following parameters:

- The event number to trigger if the rising or falling threshold exceeds the specified limit.
- The owner of the alarm.

Ensure you have configured an SNMP user and enabled SNMP notifications.

Before you begin

Make sure that you have configured an SNMP user and enabled SNMP notifications.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**rmon alarm** *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-index*] **falling-threshold** *value* [*event-index*] [**owner** *name*]
	Example:
	`switch(config)# rmon alarm 20 1.3.6.1.2.1.2.2.1.14.1 2900`
	`delta rising-threshold 1500 1 falling-threshold 0 owner test`
		Creates an RMON alarm. The value range is from –2147483647 to 2147483647. The owner name can be any alphanumeric string.

Step 3	**rmon hcalarm** *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold-high** *value* **rising-threshold-low** *value* [*event-index*] **falling-threshold-high** *value* **falling-threshold-low** *value* [*event-index*] [**owner** *name*] [**storagetype** *type*]
	Example:
	`switch(config)# rmon alarm 20 1.3.6.1.2.1.2.2.1.14.16777216 2900`
	`delta rising-threshold-high 15 rising-threshold-low 151`
	`falling-threshold-high 0 falling-threshold-low 0 owner test`
		Creates an RMON high-capacity alarm. The value range is from –2147483647 to 2147483647. The owner name can be any alphanumeric string.
		The storage type range is from 1 to 5.

Step 4	(Optional) **show rmon** {**alarms** | **hcalarms**} **Example:** `switch(config)# show rmon alarms`	Displays information about RMON alarms or high-capacity alarms.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring RMON Events

You can configure RMON events to associate with RMON alarms. You can reuse the same event with multiple RMON alarms.

Before you begin

Make sure you have configured an SNMP user and enabled SNMP notifications.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**rmon event** *index* [**description** *string*] [**log**] [**trap** *string*] [**owner** *name*] **Example:** `switch(config)# rmon event 1 trap trap1`	Configures an RMON event. The description string, trap string, and owner name can be any alphanumeric string.
Step 3	(Optional) **show rmon events** **Example:** `switch(config)# show rmon events`	Displays information about RMON events.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the RMON Configuration

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

Command	Purpose
show rmon alarms	Displays information about RMON alarms.
show rmon events	Displays information about RMON events.
show rmon hcalarms	Displays information about RMON high-capacity alarms.
show rmon logs	Displays information about RMON logs.

Configuration Examples for RMON

This example shows how to create a delta rising alarm on ifInOctets.14 and associates a notification event with this alarm:

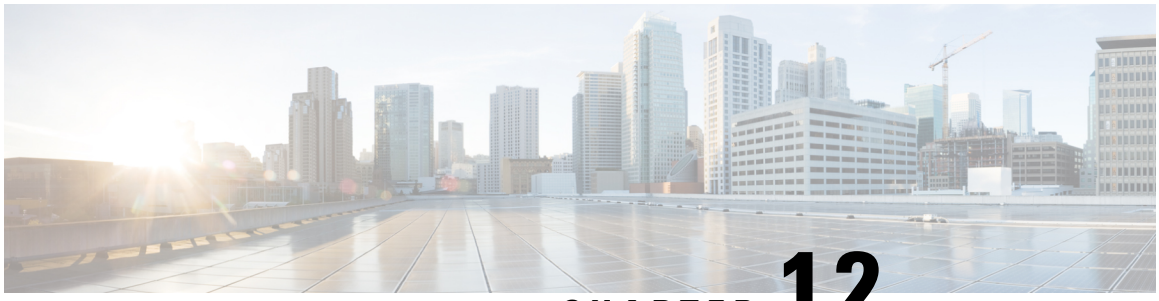
```
configure terminal
rmon alarm 20 1.3.6.1.2.1.2.2.1.14.1 2900 delta rising-threshold 1500 1 falling-threshold
0 owner test
```

```
rmon event 1 trap trap1
```

Additional References

MIBs

MIBs	MIBs Link
MIBs related to RMON	To locate and download supported MIBs, go to the following ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 12

Configuring Online Diagnostics

This chapter describes how to configure the generic online diagnostics (GOLD) feature on Cisco NX-OS devices.

This chapter contains the following sections:

About Online Diagnostics

With online diagnostics, you can test and verify the hardware functionality of the device while the device is connected to a live network.

The online diagnostics contain tests that check different hardware components and verify the data path and control signals. Disruptive online diagnostic tests (such as the disruptive loopback test) and nondisruptive online diagnostic tests (such as the ASIC register check) run during bootup, line module online insertion and removal (OIR), and system reset. The nondisruptive online diagnostic tests run as part of the background health monitoring, and you can run these tests on demand.

Online diagnostics are categorized as bootup, runtime or health-monitoring diagnostics, and on-demand diagnostics. Bootup diagnostics run during bootup, health-monitoring tests run in the background, and on-demand diagnostics run once or at user-designated intervals when the device is connected to a live network.

Bootup Diagnostics

Bootup diagnostics run during bootup and detect faulty hardware before Cisco NX-OS brings a module online. For example, if you insert a faulty module in the device, bootup diagnostics test the module and take it offline before the device uses the module to forward traffic.

Bootup diagnostics also check the connectivity between the supervisor and module hardware and the data and control paths for all the ASICs. The following table describes the bootup diagnostic tests for a module and a supervisor.

Table 11: Bootup Diagnostics

Diagnostic	Description
OBFL	Verifies the integrity of the onboard failure logging (OBFL) flash.
USB	Nondisruptive test. Checks the USB controller initialization on a module.
ManagementPortLoopback	Disruptive test, not an on-demand test. Tests loopback on the management port of a module.
EOBCPortLoopback	Disruptive test, not an on-demand test. Ethernet out of band.

Bootup diagnostics log failures to onboard failure logging (OBFL) and syslog and trigger a diagnostic LED indication (on, off, pass, or fail).

You can configure the device to either bypass the bootup diagnostics or run the complete set of bootup diagnostics.

Runtime or Health Monitoring Diagnostics

Runtime diagnostics are also called health monitoring (HM) diagnostics. These diagnostics provide information about the health of a live device. They detect runtime hardware errors, memory errors, the degradation of hardware modules over time, software faults, and resource exhaustion.

Health monitoring diagnostics are nondisruptive and run in the background to ensure the health of a device that is processing live network traffic. You can enable or disable health monitoring tests or change their runtime interval.

The following table describes the health monitoring diagnostics and test IDs for a module and a supervisor.

Table 12: Health Monitoring Nondisruptive Diagnostics

Diagnostic	Default Interval	Default Setting	Description
Module			
ACT2	30 minutes	active	Verifies the integrity of the security device on the module.
ASICRegisterCheck	1 minute	active	Checks read/write access to scratch registers for the ASICs on a module.
PrimaryBootROM	24 hours [1]	active	Verifies the integrity of the primary boot device on a module.
SecondaryBootROM	24 hours 1	active	Verifies the integrity of the secondary boot device on a module.
PortLoopback	On demand	active	Checks diagnostics on a per-port basis on all admin down ports.

Diagnostic	Default Interval	Default Setting	Description
Module			
RewriteEngineLoopback	1 minute	active	Verifies the integrity of the nondisruptive loopback for all ports up to the 1 Engine ASIC device.
AsicMemory	Only on boot up	Only on boot up - inactive	Checks if the AsicMemory is consistent using the Mbist bit in the ASIC.
FpgaRegTest	30 seconds	Health monitoring test - every 30 seconds - active	Test the FPGA status by read/write to FPGA.
Supervisor			
NVRAM	5 minutes	active	Verifies the sanity of the NVRAM blocks on a supervisor.
RealTimeClock	5 minutes	active	Verifies that the real-time clock on the supervisor is ticking.
PrimaryBootROM	30 minutes	active	Verifies the integrity of the primary boot device on the supervisor.
SecondaryBootROM	30 minutes	active	Verifies the integrity of the secondary boot device on the supervisor.
BootFlash	30 minutes	active	Verifies access to the bootflash devices.
USB	30 minutes	active	Verifies access to the USB devices.
SystemMgmtBus	30 seconds	active	Verifies the availability of the system management bus.
Mce	30 minutes	Health monitoring test - 30 minutes - active	This test uses the mcd_dameon and reports any machine check error reported by the Kernel.
Pcie	Only on boot up	Only on boot up - inactive	Reads PCIe status registers and check for any error on the PCIe device.
Console	Only on boot up	Only on boot up - inactive	This runs a port loopback test on the management port on boot up to check for its consistency.
FpgaRegTest	30 seconds	Health monitoring test - every 30 seconds - active	Test the FPGA status by read/write to FPGA.

[1] Minimum configurable test interval is 6 hours

On-Demand Diagnostics

On-demand tests help localize faults and are usually needed in one of the following situations:

- To respond to an event that has occurred, such as isolating a fault.
- In anticipation of an event that may occur, such as a resource exceeding its utilization limit.

You can run all the health monitoring tests on demand. You can schedule on-demand diagnostics to run immediately.

You can also modify the default interval for a health monitoring test.

High Availability

A key part of high availability is detecting hardware failures and taking corrective action while the device runs in a live network. Online diagnostics in high availability detect hardware failures and provide feedback to high availability software components to make switchover decisions.

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

Virtualization Support

Online diagnostics are virtual routing and forwarding (VRF) aware. You can configure online diagnostics to use a particular VRF to reach the online diagnostics SMTP server.

Guidelines and Limitations for Online Diagnostics

Online diagnostics has the following configuration guidelines and limitations:

- You cannot run disruptive online diagnostic tests on demand.
- Interface Rx and Tx packet counters are incremented (approximately four packets every 15 minutes) for ports in the shutdown state.
- On admin down ports, the unicast packet Rx and Tx counters are incremented for GOLD loopback packets. The PortLoopback test is on demand, so the packet counter is incremented only when you run the test on admin down ports.

Default Settings for Online Diagnostics

The following table lists the default settings for online diagnostic parameters.

Parameters	Default
Bootup diagnostics level	complete

Parameters	Default
Nondisruptive tests	active

Configuring Online Diagnostics

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Setting the Bootup Diagnostic Level

You can configure the bootup diagnostics to run the complete set of tests, or you can bypass all bootup diagnostic tests for a faster module bootup time.

Note We recommend that you set the bootup online diagnostics level to complete. We do not recommend bypassing the bootup online diagnostics.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**diagnostic bootup level** {**complete** | | **bypass**} **Example:** `switch(config)# diagnostic bootup level complete`	Configures the bootup diagnostic level to trigger diagnostics as follows when the device boots: • **complete**—Perform a complete set of bootup diagnostics. The default is complete. • **bypass**—Do not perform any bootup diagnostics.
Step 3	(Optional) **show diagnostic bootup level** **Example:** `switch(config)# show diagnostic bootup level`	Displays the bootup diagnostic level (bypass or complete) that is currently in place on the device.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Activating a Diagnostic Test

You can set a diagnostic test as active and optionally modify the interval (in hours, minutes, and seconds) at which the test runs.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**diagnostic monitor interval module** *slot* **test** [*test-id* | *name* | **all**] **hour** *hour* **min** *minute* **second** *second* **Example:** `switch(config)# diagnostic monitor interval module 6 test 3 hour 1 min 0 second 0`	Configures the interval at which the specified test is run. If no interval is set, the test runs at the interval set previously, or the default interval. The argument ranges are as follows: • *slot*—The range is from 1 to 10. • *test-id*—The range is from 1 to 14. • *name*—Can be any case-sensitive, alphanumeric string up to 32 characters. • *hour*—The range is from 0 to 23 hours. • *minute*—The range is from 0 to 59 minutes. • *second*—The range is from 0 to 59 seconds.
Step 3	[**no**] **diagnostic monitor module** *slot* **test** [*test-id* | *name* | **all**] **Example:** `switch(config)# diagnostic monitor interval module 6 test 3`	Activates the specified test. The argument ranges are as follows: • *slot*—The range is from 1 to 10. • *test-id*—The range is from 1 to 14. • *name*—Can be any case-sensitive, alphanumeric string up to 32 characters. The [**no**] form of this command inactivates the specified test. Inactive tests keep their current configuration but do not run at the scheduled interval.
Step 4	(Optional) **show diagnostic content module** {*slot* | **all**} **Example:** `switch(config)# show diagnostic content module 6`	Displays information about the diagnostics and their attributes.

Starting or Stopping an On-Demand Diagnostic Test

You can start or stop an on-demand diagnostic test. You can optionally modify the number of iterations to repeat this test, and the action to take if the test fails.

We recommend that you only manually start a disruptive diagnostic test during a scheduled network maintenance time.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **diagnostic ondemand iteration** *number* **Example:** `switch# diagnostic ondemand iteration 5`	Configures the number of times that the on-demand test runs. The range is from 1 to 999. The default is 1.
Step 2	(Optional) **diagnostic ondemand action-on-failure** {**continue failure-count** *num-fails* | **stop**} **Example:** `switch# diagnostic ondemand action-on-failure stop`	Configures the action to take if the on-demand test fails. The *num-fails* range is from 1 to 999. The default is 1.
Step 3	Required: **diagnostic start module** *slot* **test** [*test-id* | *name* | **all** | **non-disruptive**] [**port** *port-number* | **all**] **Example:** `switch# diagnostic start module 6 test all`	Starts one or more diagnostic tests on a module. The module slot range is from 1 to 10. The *test-id* range is from 1 to 14. The test name can be any case-sensitive, alphanumeric string up to 32 characters. The port range is from 1 to 48.
Step 4	Required: **diagnostic stop module** *slot* **test** [*test-id* | *name* | **all**] **Example:** `switch# diagnostic stop module 6 test all`	Stops one or more diagnostic tests on a module. The module slot range is from 1 to 10. The *test-id* range is from 1 to 14. The test name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 5	(Optional) **show diagnostic status module** *slot* **Example:** `switch# show diagnostic status module 6`	Verifies that the diagnostic has been scheduled.

Simulating Diagnostic Results

You can simulate a diagnostic test result.

Procedure

	Command or Action	Purpose
Step 1	**diagnostic test simulation module** *slot* **test** *test-id* {**fail** | **random-fail** | **success**} [**port** *number* | **all**] **Example:** `switch# diagnostic test simulation module 2 test 2 fail`	Simulates a test result. The *test-id* range is from 1 to 14. The port range is from 1 to 48.

Clearing Diagnostic Results

You can clear diagnostic test results.

Procedure

	Command or Action	Purpose
Step 1	**diagnostic clear result module** [*slot* | **all**] **test** {*test-id* | **all**} **Example:** `switch# diagnostic clear result module 2 test all`	Clears the test result for the specified test. The argument ranges are as follows: • *slot*—The range is from 1 to 10. • *test-id*—The range is from 1 to 14.
Step 2	**diagnostic test simulation module** *slot* **test** *test-id* clear **Example:** `switch# diagnostic test simulation module 2 test 2 clear`	Clears the simulated test result. The *test-id* range is from 1 to 14.

Verifying the Online Diagnostics Configuration

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

Command	Purpose
show diagnostic bootup level	Displays information about bootup diagnostics.
show diagnostic content module {*slot* | **all**}	Displays information about diagnostic test content for a module.
show diagnostic description module *slot* **test** [*test-name* | **all**]	Displays the diagnostic description.
show diagnostic events [**error** | **info**]	Displays diagnostic events by error and information event type.
show diagnostic ondemand setting	Displays information about on-demand diagnostics.

Command	Purpose
show diagnostic result module *slot* [**test** [*test-name* | **all**]] [**detail**]	Displays information about the results of a diagnostic.
show diagnostic simulation module *slot*	Displays information about a simulated diagnostic.
show diagnostic status module *slot*	Displays the test status for all tests on a module.
show hardware capacity [**eobc** | **forwarding** | **interface** | **module** | **power**]	Displays information about the hardware capabilities and current hardware utilization by the system.
show module	Displays module information including the online diagnostic test status.

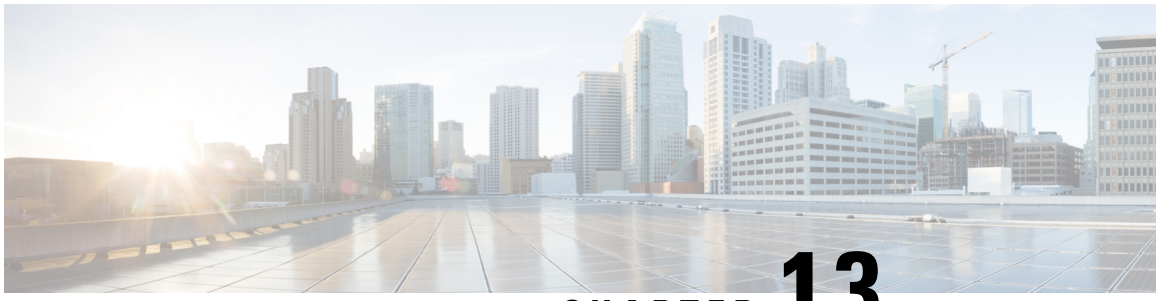
Configuration Examples for Online Diagnostics

This example shows how to start all on-demand tests on module 6:

```
diagnostic start module 6 test all
```

This example shows how to activate test 2 and set the test interval on module 6:

```
configure terminal
diagnostic monitor module 6 test 2
diagnostic monitor interval module 6 test 2 hour 3 min 30 sec 0
```



CHAPTER 13

Configuring the Embedded Event Manager

This chapter describes how to configure the Embedded Event Manager (EEM) to detect and handle critical events on Cisco NX-OS devices.

This chapter includes the following sections:

About EEM

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:

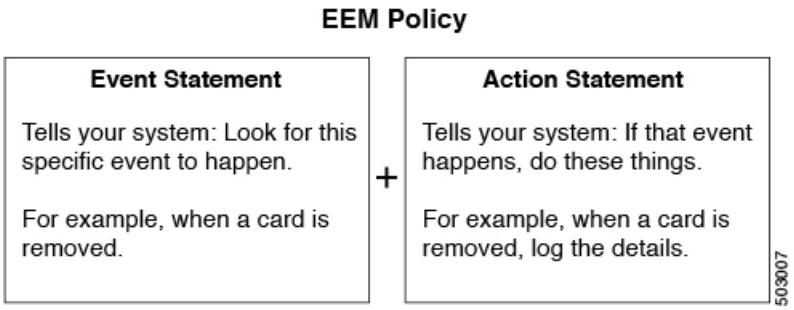
- Event statements—Events to monitor from another Cisco NX-OS component that may require some action, workaround, or notification.
- Action statements—An action that EEM can take, such as executing CLI commands, sending an email through the use of Smart Call Home feature, and disabling an interface to recover from an event.
- Policies—An event that is 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.

This figure shows the two basic statements in an EEM policy.

Figure 2: EEM Policy Statements



You can configure EEM policies using the command-line interface (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.

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.

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.

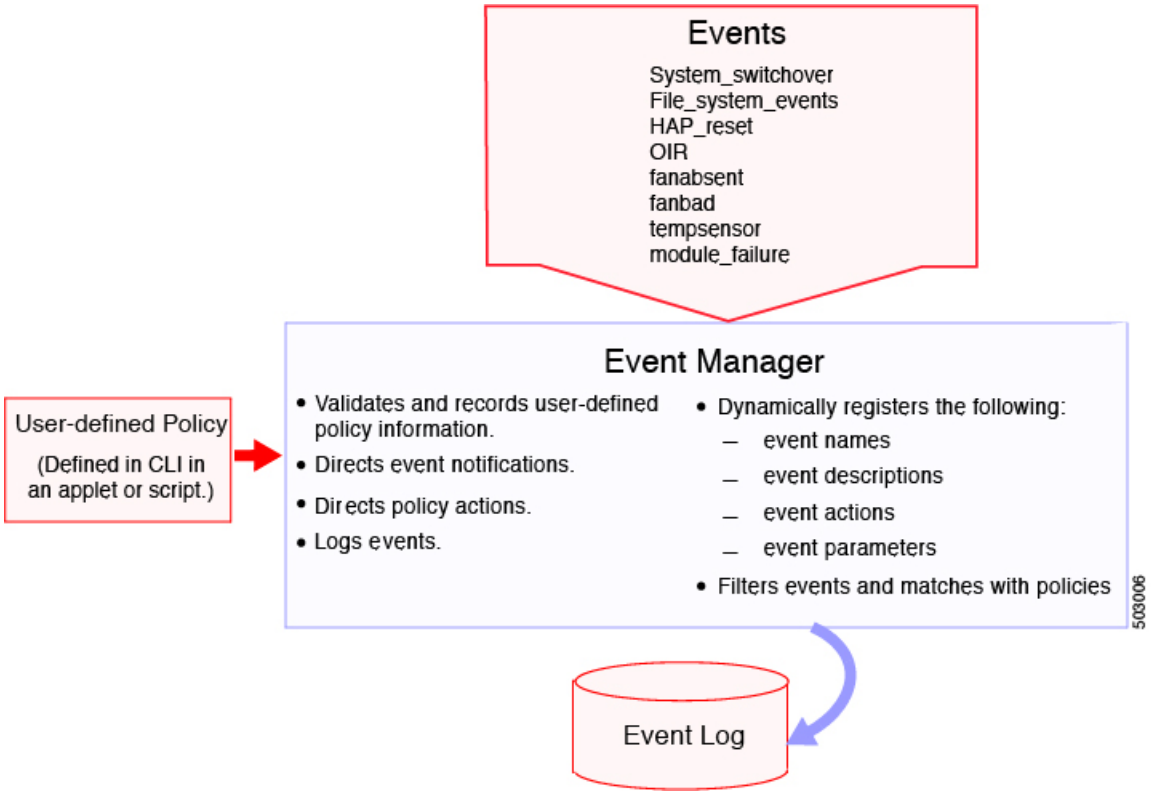
Event Statements

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.

This figure shows events that are handled by EEM.

Figure 3: EEM Overview



Event statements specify the event that triggers a policy to run. You can configure multiple event triggers.

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.

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 shutdown 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 EEM can only process a complete action cli list of up to 1024 characters in total. If more actions are required, you must define them as a new redundant applet with same trigger.

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 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 policies. After you write your VSH script policy, copy it to the device and activate it.

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 email server.

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

This example shows a sample action statement to force a module 1 shutdown, with a reset reason of "EEM action."

```
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 the following example.

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

You can reuse this environment variable in any policy.

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**), along with the count and time, you can define a combination of these events to trigger a custom action.

High Availability

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

Virtualization Support

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

Prerequisites for EEM

EEM has the following prerequisites:

- You must have network-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 will not allow the CLI command to execute.
- When you configure an EEM policy action to collect **show tech** commands, make sure to allocate enough time for the **show tech** commands to complete before the same action is called again.
- 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.
- The following rules apply to regular command expressions:
  - All regular expressions must conform to the Portable Operating System Interface for uniX (POSIX) extended standard.
  - All keywords must be expanded.
  - Only the * symbol can be used for argument replacement.

- EEM event correlation is supported only on the supervisor module.
- EEM event correlation is not supported across different modules within a single policy.
- 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, and syslog.
- 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.
- You can invoke EEM from Python. For more information about Python, see the Cisco Nexus 9000 Series NX-OS Programmability Guide.

Default Settings for EEM

This table lists the default settings for EEM parameters.

Parameters	Default
System policies	Active

Configuring EEM

You can create policies that contain actions to take based on system policies. To display information about the system policies, use the **show event manager system-policy** command.

Defining an Environment Variable

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

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**event manager environment** *variable-name variable-value* **Example:** `switch(config)# event manager environment emailto "admin@anyplace.com"`	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.

	Command or Action	Purpose
Step 3	(Optional) **show event manager environment** {*variable-name* | **all**} **Example:** `switch(config)# show event manager environment all`	Displays information about the configured environment variables.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Defining a User Policy Using the CLI

You can define a user policy using the CLI to the device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**event manager applet** *applet-name*
	Example:
	`switch(config)# event manager applet monitorShutdown`
	`switch(config-applet)#`
		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	(Optional) **description** *policy-description* **Example:** `switch(config-applet)# description "Monitors interface shutdown."`	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* **Example:** `switch(config-applet)# event cli match "conf t ; interface * ; shutdown"`	Configures the event statement for the policy. Repeat this step for multiple event statements. See Configuring Event Statements, on page 156.
Step 5	(Optional) **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*} **Example:** `switch(config-applet)# tag one or two happens 1 in 10000`	Correlates multiple events in the policy. The range for the *occurs* argument is from 1 to 4294967295. The range for the *seconds* argument is from 0 to 4294967295 seconds.

	Command or Action	Purpose
Step 6	**action** *number*[.*number2*] *action-statement* **Example:** `switch(config-applet)# action 1.0 cli show interface e 3/1`	Configures an action statement for the policy. Repeat this step for multiple action statements. See Configuring Action Statements, on page 161.
Step 7	(Optional) **show event manager policy-state** *name* [**module** *module-id*] **Example:** `switch(config-applet)# show event manager policy-state monitorShutdown`	Displays information about the status of the configured policy.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Event Statements

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

Command	Purpose
event application [**tag** *tag*] **sub-system** *sub-system-id* **type** *event-type* **Example:** `switch(config-applet)# event application sub-system 798 type 1`	Triggers an event when an event specification matches the subsystem ID and application event type. The range for the *sub-system-id* and for the *event-type* is from 1 to 4294967295. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.
event cli [**tag** *tag*] **match** *expression* [**count** *repeats* | **time** *seconds*] **Example:** `switch(config-applet)# event cli match "conf t ; interface * ; 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 *repeats* range is from 1 to 65000. The time range, in seconds, is from 0 to 4294967295, where 0 indicates no time limit.

Command	Purpose
event counter [**tag** *tag*] **name** *counter* **entry-val** *entry* **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} [**exit-val** *exit* **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 *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* **Example:** `switch(config-applet)# event fanabsent time 300`	Triggers an event if a fan is removed from the device for more than the configured time, in seconds. The *number* range is module-dependent. The *seconds* range is from 10 to 64000.
event fanbad [**fan** *number*] **time** *seconds* **Example:** `switch(config-applet)# event fanbad time 3000`	Triggers an event if a fan fails for more than the configured time, in seconds. The *number* range is module-dependent. The *seconds* range is from 10 to 64000.
event fib {**adjacency extra** | **resource tcam usage** | **route** {**extra** | **inconsistent** | **missing**}} **Example:** `switch(config-applet)# event fib adjacency extra`	Triggers an event for one of the following: • **adjacency extra**—If there is an extra route in the unicast FIB. • **resource tcam usage**—Each time the TCAM utilization percentage becomes a multiple of 5, in either direction. • **route** {**extra** | **inconsistent** | **missing**}—If a route is added, changed, or deleted in the unicast FIB.
event gold module {*slot* | **all**} **test** *test-name* [**severity** {**major** | **minor** | **moderate**}] **testing-type** {**bootup** | **monitoring** | **ondemand** | **scheduled**} **consecutive-failure** *count* **Example:** `switch(config-applet)# event gold module 2 test ASICRegisterCheck testing-type ondemand consecutive-failure 2`	Triggers an event if the named online diagnostic test experiences the configured failure severity for the configured number of consecutive failures. The *slot* range is from 1 to 10. The *test-name* is the name of a configured online diagnostic test. The *count* range is from 1 to 1000.

Command	Purpose
event interface [**tag** *tag*] {**name** *interface slot/port* **parameter**} **Example:** `switch(config-applet)# event interface ethernet 2/2 parameter`	Triggers an event if the counter is exceeded for the specified interface. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.
event memory {**critical** | **minor** | **severe**} **Example:** `switch(config-applet)# event memory critical`	Triggers an event if a memory threshold is crossed. See also Configuring Memory Thresholds, on page 165.
event module [**tag** *tag*] **status** {**online** | **offline** | **any**} **module** {**all** | *module-num*} **Example:** `switch(config-applet)# event module status offline module all`	Triggers an event if the specified module enters the selected status. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy.
event module-failure [**tag** *tag*] **type** *failure-type* **module** {*slot* | **all**} **count** *repeats* [**time** *seconds*] **Example:** `switch(config-applet)# event module-failure type lc-failed module 3 count 1`	Triggers an event if a module experiences the failure type configured. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *repeats* range is from 0 to 4294967295. The *seconds* range is from 0 to 4294967295, where 0 indicates no time limit.
event none **Example:** `switch(config-applet)# event none`	Manually runs the policy event without any events specified. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.

Command	Purpose
event oir [**tag** *tag*] {**fan** | **module** | **powersupply**} {**anyoir** | **insert** | **remove**} [*number*] **Example:** `switch(config-applet)# event oir fan remove 4`	Triggers an event if the configured device element (fan, module, or power supply) is inserted or removed from the device. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. You can optionally configure a specific fan, module, or power supply number. The *number* range is as follows: • Fan number—Module dependent. • Module number—Device dependent. • Power supply number—The range is from 1 to 3.
event policy-default count *repeats* [**time** *seconds*] **Example:** `switch(config-applet)# event policy-default count 3`	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, where 0 indicates no time limit.
event poweroverbudget **Example:** `switch(config-applet)# event poweroverbudget`	Triggers an event if the power budget exceeds the capacity of the configured power supplies.
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`	Triggers an event if the SNMP OID crosses the entry threshold based on the entry operation. 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 **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *entry* and *exit* 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.
event storm-control **Example:** `switch(config-applet)# event storm-control`	Triggers an event if traffic on a port exceeds the configured storm control threshold.

Command	Purpose
event syslog [**occurs** *count*] {**pattern** *string* | **period** *time* | **priority** *level* | **tag** *tag*} **Example:** `switch(config-applet)# event syslog period 500`	Triggers an event if the specified syslog threshold is exceeded. The range for the count is from 1 to 65000, and the range for the time is from 1 to 4294967295. The priority range is from 0 to 7. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy.
event sysmgr memory [**module** *module-num*] **major** *major-percent* **minor** *minor-percent* **clear** *clear-percent* **Example:** `switch(config-applet)# event sysmgr memory minor 80`	Triggers an event if the specified system manager memory threshold is exceeded. The range for the percentage is from 1 to 99.
event sysmgr switchover count *count* **time** *interval* **Example:** `switch(config-applet)# event sysmgr switchover count 10 time 1000`	Triggers an event if the specified switchover count is exceeded within the time interval specified. The switchover count is from 1 to 65000. The time interval is from 0 to 2147483647.
event temperature [**module** *slot*] [*sensor-number*] **threshold** {**any** | **major** | **minor**} **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.

Command	Purpose
event timer {**absolute time** *time* **name** *name* | **countdown time** *time* **name** *name* | **cron cronentry** *string* | **tag** *tag* | **watchdog time** *time* **name** *name*} **Example:** `switch(config-applet)# event timer absolute time 100 name abtimer`	Triggers an event if the specified time is reached. The range for the time is from 1 to 4294967295. • **absolute time**—Triggers an event when the specified absolute time of day occurs. • **countdown time**—Triggers an event when when the specified time counts down to zero. The timer does not reset. • **cron cronentry**—Triggers an event when the CRON string specification matches the current time. • **watchdog time**—Triggers an event when the specified time counts down to zero. The timer automatically resets to the initial value and continues to count down. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.
event track [**tag** *tag*] *object-number* **state** {**any** | **down** | **up**} **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 *object-number* range is from 1 to 500.

Configuring Action Statements

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

Command	Purpose
action *number*[*.number2*] **cli** *command1* [*command2...*] [**local**] **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 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*] **counter name** *counter* **value** *val* **op** {**dec** | **inc** | **nop** | **set**} **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 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** **Example:** `switch(config-applet)# action 1.0 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*] **forceshut** [**module** *slot* | **xbar** *xbar-number*] **reset-reason** *seconds* **Example:** `switch(config-applet)# action 1.0 forceshut module 2 reset-reason "flapping links"`	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 reset reason is a quoted alphanumeric string up to 80 characters.
action *number*[*.number2*] **overbudgetshut** [**module** *slot*[*-slot*]] **Example:** `switch(config-applet)# action 1.0 overbudgetshut module 3-5`	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.
action *number*[*.number2*] **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 number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[*.number2*] **publish-event** **Example:** `switch(config-applet)# action 1.0 publish-event`	Forces the publication of an application-specific 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*] **reload** [**module** *slot*[*-slot*]] **Example:** `switch(config-applet)# action 1.0 reload module 3-5`	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.

Command	Purpose
action *number*[*.number2*] **snmp-trap** {[**intdata1** *data* [**intdata2** *data*]] [**strdata** *string*]} **Example:** `switch(config-applet)# action 1.0 snmp-trap strdata "temperature problem"`	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* **Example:** `switch(config-applet)# action 1.0 syslog priority notifications msg "cpu high"`	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 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 will 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

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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**event manager policy** *policy-script* **Example:** `switch(config)# event manager policy moduleScript`	Registers and activates an EEM script policy. The *policy-script* can be any case-sensitive alphanumeric string up to 29 characters.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Overriding a Policy

You can override a system policy.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show event manager policy-state** *system-policy*
	Example:
	`switch(config-applet)# 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`
		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*
	Example:
	`switch(config)# event manager applet ethport override __ethpm_link_flap`
	`switch(config-applet)#`
		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.

	Command or Action	Purpose
Step 4	(Optional) **description** *policy-description* **Example:** `description "Overrides link flap policy."`	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	Required: **event** *event-statement* **Example:** `switch(config-applet)# event policy-default count 2 time 1000`	Configures the event statement for the policy.
Step 6	Required: **action** *number action-statement* **Example:** `switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."`	Configures an action statement for the policy. Repeat this step for multiple action statements.
Step 7	(Optional) **show event manager policy-state** *name* **Example:** `switch(config-applet)# show event manager policy-state ethport`	Displays information about the configured policy.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Memory Thresholds

You can set the memory thresholds that are used to trigger events and set whether the operating system should kill processes if it cannot allocate memory.

Before you begin

Ensure that you are logged in with administrator privileges.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**system memory-thresholds minor** *minor* **severe** *severe* **critical** *critical* **Example:** `switch(config)# system memory-thresholds minor 60 severe 70 critical 80`	Configures the system memory thresholds that generate EEM memory events. The default values are as follows: • Minor-85 • Severe-90 • Critical-95 When these memory thresholds are exceeded, the system generates the following syslogs: • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE • 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR ALERT RECOVERED • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE ALERT RECOVERED • 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL ALERT RECOVERED
Step 3	(Optional) **system memory-thresholds threshold critical no-process-kill** **Example:** `switch(config)# system memory-thresholds threshold critical no-process-kill`	Configures the system to not kill processes when the memory cannot be allocated. The default value is to allow the system to kill processes, starting with the one that consumes the most memory.
Step 4	(Optional) **show running-config | include "system memory"** **Example:** `switch(config-applet)# show running-config | include "system memory"`	Displays information about the system memory configuration.

	Command or Action	Purpose
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Syslog as 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

EEM should be available for registration by syslog.

The syslog daemon must be configured and executed.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**event manager applet** *applet-name*
	Example:
	`switch(config)# event manager applet abc`
	`switch(config-applet)#`
		Registers an applet with EEM and enters applet configuration mode.

Step 3	**event syslog** [**tag** *tag*] {**occurs** *number* | **period** *seconds* | **pattern** *msg-text* | **priority** *priority*} **Example:** `switch(config-applet)# event syslog occurs 10`	Monitors syslog messages and invokes the policy based on the search string in the policy. • The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. • The **occurs** *number* keyword-argument pair specifies the number of occurrences. The range is from 1 to 65000. • The **period** *seconds* keyword-argument pair specifies the interval during which the event occurs. The range is from 1 to 4294967295.

	Command or Action	Purpose
		• The **pattern** *msg-text* 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** *priority* 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	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the 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-state *policy-name*	Displays information about the 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.

Configuration Examples for EEM

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

```
event manager applet ethport override __ethpm_link_flap
event policy-default count 2 time 1000
action 1 cli conf t
action 2 cli int et1/1
action 3 cli no shut
```

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:

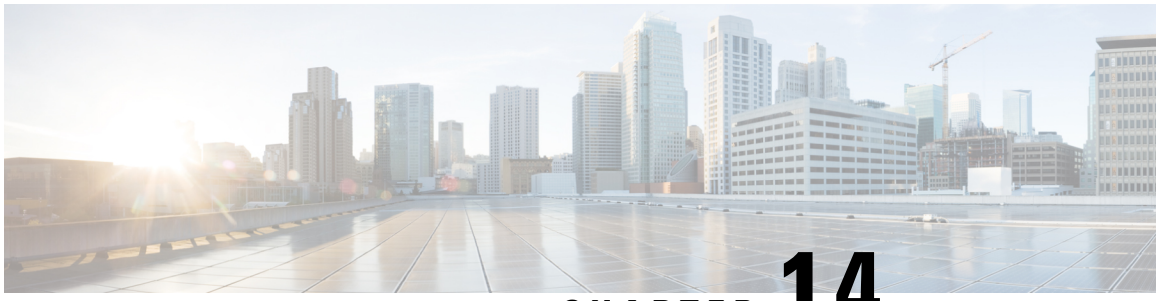
```
event manager applet TEST
event cli match "conf t"
action 1.0 snmp-trap strdata "Configuration change"
action 2.0 event-default
```

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

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.

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

Note For additional EEM configuration examples, see .



CHAPTER 14

Configuring Onboard Failure Logging

This chapter describes how to configure the onboard failure logging (OBFL) features on Cisco NX-OS devices.

This chapter includes the following sections:

About OBFL

Cisco NX-OS provides the ability to log failure data to persistent storage, which you can retrieve and display for analysis at a later time. This onboard failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. The information will help analyze failed modules.

OBFL stores the following types of data:

- Time of initial power-on
- Slot number of the module in the chassis
- Initial temperature of the module
- Firmware, BIOS, FPGA, and ASIC versions
- Serial number of the module
- Stack trace for crashes
- CPU hog information
- Memory leak information
- Software error messages
- Hardware exception logs

- Environmental history
- OBFL-specific history information
- ASIC interrupt and error statistics history
- ASIC register dumps

Prerequisites for OBFL

You must have network-admin user privileges.

Guidelines and Limitations for OBFL

OBFL has the following guidelines and limitations:

- OBFL is enabled by default.
- OBFL flash supports a limited number of writes and erases. The more logging you enable, the faster you use up this number of writes and erases.

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Default Settings for OBFL

The following table lists the default settings for OBFL parameters.

Parameters	Default
OBFL	All features enabled

Configuring OBFL

You can configure the OBFL features on Cisco NX-OS devices.

Before you begin

Make sure that you are in global configuration mode.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hw-module logging onboard**
	Example:
	`switch(config)# hw-module logging onboard`
	`Module: 7 Enabling ... was successful.`
	`Module: 10 Enabling ... was successful.`
	`Module: 12 Enabling ... was successful.`
		Enables all OBFL features.

Step 3	**hw-module logging onboard counter-stats**
	Example:
	`switch(config)# hw-module logging onboard counter-stats`
	`Module: 7 Enabling counter-stats ... was successful.`
	`Module: 10 Enabling counter-stats ... was successful.`
	`Module: 12 Enabling counter-stats ... was successful.`
		Enables the OBFL counter statistics.

Step 4	**hw-module logging onboard cpuhog**
	Example:
	`switch(config)# hw-module logging onboard cpuhog`
	`Module: 7 Enabling cpu-hog ... was successful.`
	`Module: 10 Enabling cpu-hog ... was successful.`
	`Module: 12 Enabling cpu-hog ... was successful.`
		Enables the OBFL CPU hog events.

Step 5	**hw-module logging onboard environmental-history**
	Example:
	`switch(config)# hw-module logging onboard environmental-history`
	`Module: 7 Enabling environmental-history ... was successful.`
	`Module: 10 Enabling environmental-history ... was successful.`
	`Module: 12 Enabling environmental-history ... was successful.`
		Enables the OBFL environmental history.

Step 6	**hw-module logging onboard error-stats**	Enables the OBFL error statistics.

	Command or Action	Purpose

	Example:
	`switch(config)# hw-module logging onboard error-stats`
	`Module: 7 Enabling error-stats ... was successful.`
	`Module: 10 Enabling error-stats ... was successful.`
	`Module: 12 Enabling error-stats ... was successful.`	

Step 7	**hw-module logging onboard interrupt-stats**
	Example:
	`switch(config)# hw-module logging onboard interrupt-stats`
	`Module: 7 Enabling interrupt-stats ... was successful.`
	`Module: 10 Enabling interrupt-stats ... was successful.`
	`Module: 12 Enabling interrupt-stats ... was successful.`
		Enables the OBFL interrupt statistics.

Step 8	**hw-module logging onboard module** *slot*
	Example:
	`switch(config)# hw-module logging onboard module 7`
	`Module: 7 Enabling ... was successful.`
		Enables the OBFL information for a module.

Step 9	**hw-module logging onboard obfl-logs**
	Example:
	`switch(config)# hw-module logging onboard obfl-logs`
	`Module: 7 Enabling obfl-log ... was successful.`
	`Module: 10 Enabling obfl-log ... was successful.`
	`Module: 12 Enabling obfl-log ... was successful.`
		Enables the boot uptime, device version, and OBFL history.

Step 10	(Optional) **show logging onboard** **Example:** `switch(config)# show logging onboard`	Displays information about OBFL. **Note** To display OBFL information stored in flash on a module, see Verifying the OBFL Configuration, on page 175.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the OBFL Configuration

To display OBFL information stored in flash on a module, perform one of the following tasks:

Command	Purpose
show logging onboard boot-uptime	Displays the boot and uptime information.
show logging onboard counter-stats	Displays statistics on all ASIC counters.
show logging onboard credit-loss	Displays OBFL credit loss logs.
show logging onboard device-version	Displays device version information.
show logging onboard endtime	Displays OBFL logs to a specified end time.
show logging onboard environmental-history	Displays environmental history.
show logging onboard error-stats	Displays error statistics.
show logging onboard exception-log	Displays exception log information.
show logging onboard interrupt-stats	Displays interrupt statistics.
show logging onboard module *slot*	Displays OBFL information for a specific module.
show logging onboard obfl-history	Displays history information.
show logging onboard obfl-logs	Displays log information.
show logging onboard stack-trace	Displays kernel stack trace information.
show logging onboard starttime	Displays OBFL logs from a specified start time.
show logging onboard status	Displays OBFL status information.

Use the **show logging onboard status** command to display the configuration status of OBFL.

```
switch# show logging onboard status
----------------------------
OBFL Status
----------------------------
Switch OBFL Log: Enabled

Module: 4 OBFL Log: Enabled
cpu-hog Enabled
credit-loss Enabled
environmental-history Enabled
error-stats Enabled
exception-log Enabled
interrupt-stats Enabled
mem-leak Enabled
miscellaneous-error Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
register-log Enabled
request-timeout Enabled
stack-trace Enabled
system-health Enabled
timeout-drops Enabled
stack-trace Enabled

Module: 22 OBFL Log: Enabled
```

```
cpu-hog Enabled
credit-loss Enabled
environmental-history Enabled
error-stats Enabled
exception-log Enabled
interrupt-stats Enabled
mem-leak Enabled
miscellaneous-error Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
register-log Enabled
request-timeout Enabled
stack-trace Enabled
system-health Enabled
timeout-drops Enabled
stack-trace Enabled
```

Use the **clear logging onboard** command to clear the OBFL information for each of the **show** command options listed.

Configuration Example for OBFL

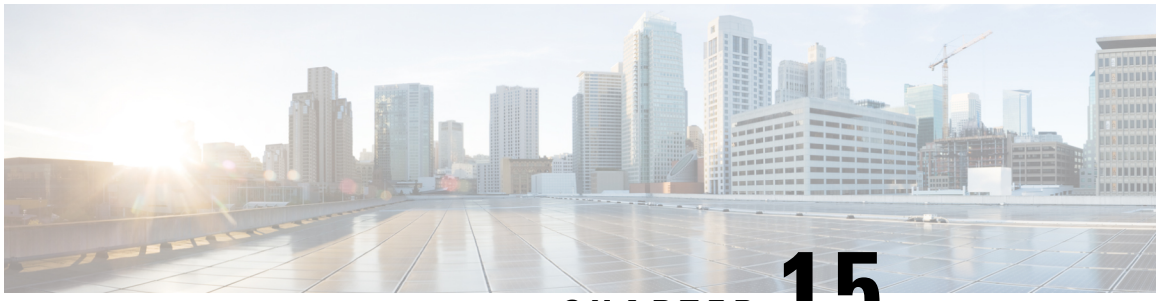
This example shows how to enable OBFL on module 2 for environmental information:

```
switch# configure terminal
switch(config)# hw-module logging onboard module 2 environmental-history
```

Additional References

Related Documents

Related Topic	Document Title
Configuration files	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 15

Configuring SPAN

This chapter describes how to configure an Ethernet switched port analyzer (SPAN) to analyze traffic between ports on Cisco NX-OS devices.

About SPAN

SPAN analyzes all traffic between source ports by directing the SPAN session traffic to a destination port with an external analyzer attached to it.

You can define the sources and destinations to monitor in a SPAN session on the local device.

SPAN Sources

The interfaces from which traffic can be monitored are called SPAN sources. Sources designate the traffic to monitor and whether to copy ingress (Rx), egress (Tx), or both directions of traffic. SPAN sources include the following:

- Ethernet ports (but not subinterfaces)

Characteristics of Source Ports

SPAN source ports have the following characteristics:

- A port configured as a source port cannot also be configured as a destination port.

SPAN Destinations

SPAN destinations refer to the interfaces that monitor source ports. Destination ports receive the copied traffic from SPAN sources. SPAN destinations include the following:

- Ethernet ports in either access or trunk mode

Characteristics of Destination Ports

SPAN destination ports have the following characteristics:

- A port configured as a destination port cannot also be configured as a source port.
- A destination port can be configured in only one SPAN session at a time.
- Destination ports do not participate in any spanning tree instance. SPAN output includes bridge protocol data unit (BPDU) Spanning Tree Protocol hello packets.

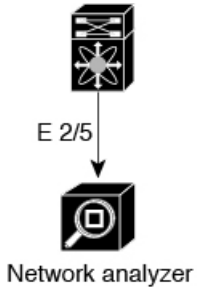
SPAN Sessions

You can create SPAN sessions to designate sources and destinations to monitor.

See the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* for information on the number of supported SPAN sessions.

This figure shows a SPAN configuration. Packets on three Ethernet ports are copied to destination port Ethernet 2/5. Only traffic in the direction specified is copied.

Figure 4: SPAN Configuration



Source Port	Direction	Destination Ports
E 2/1	Rx	E 2/5
E 2/2	Rx, Tx	
E 2/3	Tx	

503005

Localized SPAN Sessions

A SPAN session is localized when all of the source interfaces are on the same line card. A session destination interface can be on any line card.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware. For information on the TCAM regions used by SPAN sessions, see the "Configuring IP ACLs" chapter of the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

High Availability

The SPAN feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied. For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.

Prerequisites for SPAN

SPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired SPAN configuration. For more information, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

Guidelines and Limitations for SPAN

Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

SPAN has the following configuration guidelines and limitations:

- Traffic that is denied by an ACL may still reach the SPAN destination port because SPAN replication is performed on the ingress side prior to the ACL enforcement (ACL dropping traffic).
- For SPAN session limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- All SPAN replication is performed in the hardware. The supervisor CPU is not involved.
- You can configure a SPAN session on the local device only. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- Packets with FCS errors are not mirrored in a SPAN session.
- FEX and SPAN port-channel destinations are not supported on the Cisco Nexus 9500 platform switches with an -EX or –FX type line card.
- You can configure only one destination port in a SPAN session.
- A destination port can be configured in only one SPAN session at a time.
- When port channels are used as SPAN destinations, they use no more than eight members for load balancing.
- SPAN does not support destinations on Cisco Nexus 9408PC-CFP2 line card ports.
- Configuring two SPAN or ERSPAN sessions on the same source interface with only one filter is not supported. If the same source is used in multiple SPAN or ERSPAN sessions either all the sessions must have different filters or no sessions should have filters.
- The following guidelines apply to SPAN copies of access port dot1q headers:
  - When traffic ingresses from a trunk port and egresses to an access port, an egress SPAN copy of an access port on a switch interface always has a dot1q header.

  - When traffic ingresses from an access port and egresses to a trunk port, an ingress SPAN copy of an access port on a switch interface does not have a dot1q header.
  - When traffic ingresses from an access port and egresses to an access port, an ingress/egress SPAN copy of an access port on a switch interface does not have a dot1q header.

- You cannot configure a port as both a source and destination port.
- Enabling UniDirectional Link Detection (UDLD) on the SPAN source and destination ports simultaneously is not supported. If UDLD frames are expected to be captured on the source port of such SPAN session, disable UDLD on the destination port of the SPAN session.
- SPAN is not supported for management ports.
- Statistics are not support for the filter access group.
- SPAN is supported in Layer 3 mode; however, SPAN is not supported on Layer 3 subinterfaces or Layer 3 port-channel subinterfaces.
- When a SPAN session contains source ports that are monitored in the transmit or transmit and receive direction, packets that these ports receive might be replicated to the SPAN destination port even though the packets are not actually transmitted on the source ports. Some examples of this behavior on source ports are as follows:
  - Traffic that results from flooding
  - Broadcast and multicast traffic
- SPAN sessions cannot capture packets with broadcast or multicast MAC addresses that reach the supervisor, such as ARP requests and Open Shortest Path First (OSPF) protocol hello packets, if the source of the session is the supervisor Ethernet in-band interface. To capture these packets, you must use the physical interface as the source in the SPAN sessions.
- VLAN SPAN monitors only the traffic that enters Layer 2 ports in the VLAN.
- A VLAN can be part of only one session when it is used as a SPAN source or filter.
- VLAN ACL redirects to SPAN destination ports are not supported.
- When using a VLAN ACL to filter a SPAN, only **action forward** is supported; **action drop** and **action redirect** are not supported.
- For VXLAN/VTEP, SPAN source or destination is supported on any port.
- The number of SPAN sessions per line card reduces to two if the same interface is configured as a bidirectional source in more than one session. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- A single forwarding engine instance supports four SPAN sessions. For Cisco Nexus 9300 Series switches, if the first three sessions have bidirectional sources, the fourth session has hardware resources only for Rx sources. This limitation might also apply to Cisco Nexus 9500 Series switches, depending on the SPAN source's forwarding engine instance mappings. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- An access-group filter in a SPAN session must be configured as vlan-accessmap. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.

- Supervisor-generated stream of bytes module header (SOBMH) packets have all of the information to go out on an interface and can bypass all forwarding lookups in the hardware, including SPAN and ERSPAN. CPU-generated frames for Layer 3 interfaces and the Bridge Protocol Data Unit (BPDU) class of packets are sent using SOBMH. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards. The Cisco Nexus 9636C-R and 9636Q-R both support inband SPAN and local SPAN.
- IPv6 ACL filters for Layer 2 ports are not supported on Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q switch.
- Cisco NX-OS does not span Link Layer Discovery Protocol (LLDP) or Link Aggregation Control Protocol (LACP) packets when the source interface is not a host interface port channel.

The following guidelines and limitations apply to egress (Tx) SPAN:

- The following limitations apply to egress (Tx) SPAN and these switches:
  - Cisco Nexus 92160YC-X
  - Cisco Nexus 92304QC
  - Cisco Nexus 9272Q
  - Cisco Nexus 9236C
  - Cisco Nexus 92300YC

  ACL filtering is not supported (applies to both unicast and Broadcast, Unknown Unicast and Multicast (BUM) traffic)

  VLAN filtering is supported, but only for unicast traffic

  VLAN filtering is not supported for BUM traffic
- SPAN copies for multicast packets are made prior to rewrite. Therefore, the TTL, VLAN ID, any remarking due to egress policy, and so on, are not captured in the SPAN copy.
- If SPAN is mirroring the traffic which ingresses on an interface in an ASIC instance and egresses on a Layer 3 interface (SPAN Source) on a different ASIC instance, then TX mirrored packet will have a VLAN ID 4095 on Cisco Nexus 9500 platform modular switches using non-EX line cards.
- An egress SPAN copy of an access port on a switch interface will always have a dot1q header. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- The flows for post-routed unknown unicast flooded packets are in the SPAN session, even if the SPAN session is configured to not monitor the ports on which this flow is forwarded. This limitation applies to Network Forwarding Engine (NFE) and NFE2-enabled EOR switches and SPAN sessions that have Tx port sources.
- Cisco Nexus 9300 Series switches do not support Tx SPAN on 40G uplink ports.

Note This limitation does not apply to Nexus 9300-EX/FX/FX2 platform switches that have the 100G interfaces.

The following guidelines and limitations apply to ingress (Rx) SPAN:

- A SPAN copy of Cisco Nexus 9300 Series switch 40G uplink interfaces will miss the dot1q information when spanned in the Rx direction.

Note This limitation does not apply to Nexus 9300-EX/FX/FX2 platform switches that have the 100G interfaces.

- Session filtering functionality (VLAN or ACL filters) is supported only for Rx sources. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.

The following guidelines and limitations apply to FEX ports:

- The FEX NIF interfaces or port-channels cannot be used as a SPAN source or SPAN destination. If the FEX NIF interfaces or port-channels are specified as a SPAN source or SPAN destination, the software displays an unsupported error.
- Cisco Nexus 9300 and 9500 platform switches support FEX ports as SPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic flows through the switch and FEX. Routed traffic might not be seen on FEX HIF egress SPAN.
- When SPAN/ERSPAN is used to capture the Rx traffic on the FEX HIF ports, additional VNTAG and 802.1q tags are present in the captured traffic.
- VLAN and ACL filters are not supported for FEX ports.
- If the sources used in bidirectional SPAN sessions are from the same FEX, the hardware resources are limited to two SPAN sessions.

The following guidelines and limitations apply to Cisco Nexus 9200 and 9300-EX Series switches:

The following guidelines and limitations apply to SPAN truncation:

- Truncation is supported only for local and SPAN source sessions. It is not supported for SPAN destination sessions.
- Configuring MTU on a SPAN session truncates all of the packets egressing on the SPAN destination (for that session) to the MTU value specified.
- The cyclic redundancy check (CRC) is recalculated for the truncated packet.
- The bytes specified are retained starting from the header of the packets. The rest are truncated if the packet is longer than the MTU.

Default Settings for SPAN

The following table lists the default settings for SPAN parameters.

Parameters	Default
SPAN sessions	Created in the shut state

Configuring SPAN

Note Cisco NX-OS commands for this feature may differ from those in Cisco IOS.

Configuring a SPAN Session

You can configure a SPAN session on the local device only. By default, SPAN sessions are created in the shut state.

Note For bidirectional traditional sessions, you can configure the sessions without specifying the direction of the traffic.

Before you begin

You must configure the destination ports in access or trunk mode. For more information, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *ethernet slot/port*
	Example:
	`switch(config)# interface ethernet 2/5`
	`switch(config-if)#`
		Enters interface configuration mode on the selected slot and port.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures switchport parameters for the selected slot and port or range of ports.
Step 4	**switchport monitor** **Example:** `switch(config-if)# switchport monitor`	Configures the switchport interface as a SPAN destination.
Step 5	(Optional) Repeat Steps 2 through 4 to configure monitoring on additional SPAN destinations.	—

	Command or Action	Purpose
Step 6	**no monitor session** *session-number* **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified SPAN session. The new session configuration is added to the existing session configuration.

Step 7	**monitor session** *session-number* [**shut**]
	Example:
	Example:
	`switch(config)# monitor session 3 shut`
	`switch(config-monitor)#`
		Enters the monitor configuration mode. The new session configuration is added to the existing session configuration. By default, the session is created in the shut state, and the session is a local SPAN session. The optional keyword shut specifies a shut state for the selected session.

Step 8	**description** *description* **Example:** `switch(config-monitor)# description my_span_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

Step 9	**source** {**interface** *type* [**rx** | **tx** | **both**] | [**rx**]}
	Example:
	`switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx`
	Example:
	`switch(config-monitor)# source interface port-channel 2`
		You can configure one or more sources, as either a series of comma-separated entries or a range of numbers.
		You can specify the traffic direction to copy as ingress (rx), egress (tx), or both.
		For a unidirectional session, the direction of the source must match the direction specified in the session.

Step 10	(Optional) **filter access-group** *acl-filter* **Example:** `switch(config-monitor)# filter access-group ACL1`	Associates an ACL with the SPAN session.
Step 11	Required: **destination interface** *type slot*/*port* **Example:** `switch(config-monitor)# destination interface ethernet 2/5`	Configures a destination for copied source packets. **Note** The SPAN destination port must be either an access port or a trunk port. **Note** You must enable monitor mode on the destination port.
Step 12	Required: **no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 13	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:**	Displays the SPAN configuration.

	Command or Action	Purpose
	`switch(config-monitor)# show monitor session 3`	
Step 14	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Shutting Down or Resuming a SPAN Session

You can shut down SPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, SPAN sessions are created in the shut state.

You can resume (enable) SPAN sessions to resume the copying of packets from sources to destinations. In order to enable a SPAN session that is already enabled but operationally down, you must first shut it down and then enable it.

You can configure the shut and enabled SPAN session states with either a global or monitor configuration mode command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# monitor session 3 shut`	Shuts down the specified SPAN sessions. By default, sessions are created in the shut state. The **no** form of the command resumes (enables) the specified SPAN sessions. By default, sessions are created in the shut state. **Note** If a monitor session is enabled but its operational status is down, to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.

Step 3	**monitor session** *session-number*
	Example:
	`switch(config)# monitor session 3`
	`switch(config-monitor)#`
		Enters the monitor configuration mode. The new session configuration is added to the existing session configuration.

	Command or Action	Purpose
Step 4	[**no**] **shut** **Example:** `switch(config-monitor)# shut`	Shuts down the SPAN session. By default, the session is created in the shut state. The **no** form of the command enables the SPAN session. By default, the session is created in the shut state.
Step 5	(Optional) **show monitor** **Example:** `switch(config-monitor)# show monitor`	Displays the status of SPAN sessions.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the SPAN Configuration

To display the SPAN configuration, perform one of the following tasks:

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the SPAN session configuration.

Configuration Examples for SPAN

Configuration Example for a SPAN Session

To configure a SPAN session, follow these steps:

Procedure

Step 1 Configure destination ports in access mode and enable SPAN monitoring.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Step 2 Configure a SPAN session.

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-monitor)# source interface port-channel 2
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Configuration Example for a Unidirectional SPAN Session

To configure a unidirectional SPAN session, follow these steps:

Procedure

Step 1 Configure destination ports in access mode and enable SPAN monitoring.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Step 2 Configure a SPAN session.

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3 rx
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Configuration Example for a SPAN ACL

This example shows how to configure a SPAN ACL:

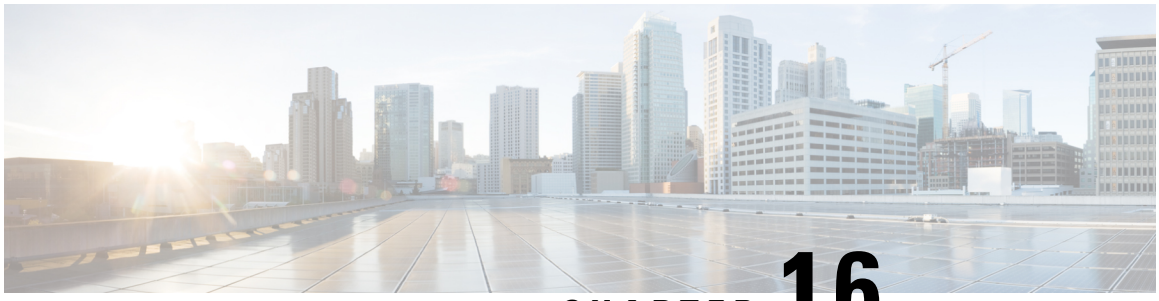
```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
```

```
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config-access-map)# match ip address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config-access-map)# match ip address match_12_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1
switch(config-erspan-src)# filter access_group span_filter
```

Additional References

Related Documents

Related Topic	Document Title
FEX	*Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 9000 Series Switches*



CHAPTER 16

Configuring ERSPAN

This chapter describes how to configure an encapsulated remote switched port analyzer (ERSPAN) to transport mirrored traffic in an IP network on Cisco NX-OS devices.

This chapter contains the following sections:

About ERSPAN

ERSPAN transports mirrored traffic over an IP network, which provides remote monitoring of multiple switches across your network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface.

ERSPAN Types

Cisco Nexus 9300 Series switches support ERSPAN Type II, and Cisco Nexus 9500 Series switches support only ERSPAN.

ERSPAN Sources

The interfaces from which traffic can be monitored are called ERSPAN sources. Sources designate the traffic to monitor and whether to copy ingress, egress, or both directions of traffic. ERSPAN sources include the following:

- Ethernet ports (but not subinterfaces)
- Forward drops

Note A single ERSPAN session can include mixed sources in any combination of the above.

ERSPAN Sessions

You can create ERSPAN sessions that designate sources to monitor.

Localized ERSPAN Sessions

An ERSPAN session is localized when all of the source interfaces are on the same line card.

High Availability

The ERSPAN feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied.

For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.

Prerequisites for ERSPAN

ERSPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired ERSPAN configuration. For more information, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

Guidelines and Limitations for ERSPAN

Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

ERSPAN has the following configuration guidelines and limitations:

- ERSPAN destination handles jumbo frames for MTU differently based on the platform. For the following Cisco Nexus 9300 platform switches (and supporting line cards), ERSPAN destination drops the jumbo frames:

  Switches

  - Cisco Nexus 9332PQ
  - Cisco Nexus 9372PX
  - Cisco Nexus 9372PX-E
  - Cisco Nexus 9372TX
  - Cisco Nexus 9372TX-E

- Cisco Nexus 93120TX

Line Cards

- Cisco Nexus 9564PX
- Cisco Nexus 9464TX
- Cisco Nexus 9464TX2
- Cisco Nexus 9564TX
- Cisco Nexus 9464PX
- Cisco Nexus 9536PQ
- Cisco Nexus 9636PQ
- Cisco Nexus 9432PQ

For the following Cisco Nexus 9200-series switches (and supporting line cards), ERSPAN truncates the packets at port MTU, and issues a TX Output error:

Switches

- Cisco Nexus 92160YC-X
- Cisco Nexus 92304QC
- Cisco Nexus 9272Q
- Cisco Nexus 9232C
- Cisco Nexus 9236C
- Cisco Nexus 92300YC
- Cisco Nexus 93108TC-EX
- Cisco Nexus 93180LC-EX
- Cisco Nexus 93180YC-EX

Line Cards

- Cisco Nexus 9736C-EX
- Cisco Nexus 97160YC-EX
- Cisco Nexus 9732C-EX
- Cisco Nexus 9732C-EXM

- For ERSPAN session limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- The number of ERSPAN sessions per line card reduces to two if the same interface is configured as a bidirectional source in more than one session.
- Only ERSPAN source sessions are supported. Destination sessions are not supported.

Note Support for destination sessions on Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches is available in Cisco NX-OS Release 9.3(1). See the Configuring ERSPAN chapter in the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.3(x)* for more information.

- Configuring two SPAN or ERSPAN sessions on the same source interface with only one filter is not supported. If the same source is used in multiple SPAN or ERSPAN sessions either all the sessions must have different filters or no sessions should have filters.
- Packets with FCS errors are not mirrored in an ERSPAN session.
- TCAM carving is not required for SPAN/ERSPAN on the following line cards:
  - Cisco Nexus 9636C-R
  - Cisco Nexus 9636Q-R
  - Cisco Nexus 9636C-RX
  - Cisco Nexus 96136YC-R

Note All other switches supporting SPAN/ERSPAN must use TCAM carving.

- Statistics are not supported for the filter access group.
- An access-group filter in an ERSPAN session must be configured as vlan-accessmap.
- All ERSPAN replication is performed in the hardware. The supervisor CPU is not involved.
- Control plane packets generated by the supervisor cannot be ERSPAN encapsulated or filtered by an ERSPAN access control list (ACL).
- ERSPAN is not supported for management ports.
- ERSPAN does not support destinations on Layer 3 port-channel subinterfaces.
- ERSPAN and ERSPAN ACL sessions are terminated identically at the destination router only when the ERSPAN destination IP address is resolved through Cisco Nexus 9300 platform switch uplink ports.
- Supervisor-generated stream of bytes module header (SOBMH) packets have all of the information to go out on an interface and can bypass all forwarding lookups in the hardware, including SPAN and ERSPAN. CPU-generated frames for Layer 3 interfaces and the Bridge Protocol Data Unit (BPDU) class of packets are sent using SOBMH. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards. The Cisco Nexus 9636C-R and 9636Q-R line cards both support inband SPAN and local SPAN.
- A VLAN can be part of only one session when it is used as an ERSPAN source or filter.
- VLAN ERSPAN monitors only the traffic that leaves or enters Layer 2 ports in the VLAN.
- If you enable ERSPAN on a vPC and ERSPAN packets need to be routed to the destination through the vPC, packets that come through the vPC peer link cannot be captured.

- ERSPAN is not supported over a VXLAN overlay.
- ERSPAN copies for multicast packets are made prior to rewrite. Therefore, the TTL, VLAN ID, any remarking due to egress policy, and so on are not captured in the ERSPAN copy.
- Marker packet for ERSPAN is not supported on Cisco Nexus 9508 switches with an 9732C-EX line card.
- Cisco Nexus 9300-EX/FX switches cannot serve as an ERSPAN destination for Cisco Nexus 3000 and non-EX/FX Cisco Nexus 9000 switches.

The following guidelines and limitations apply to egress (Tx) ERSPAN:

- Cisco Nexus 9300 Series switches do not support Tx ERSPAN on 40G uplink ports.
- The flows for post-routed unknown unicast flooded packets are in the ERSPAN session, even if the ERSPAN session is configured to not monitor the ports on which this flow is forwarded. This limitation applies to Network Forwarding Engine (NFE) and NFE2-enabled EOR switches and ERSPAN sessions that have TX port sources.

The following guidelines and limitations apply to ingress (Rx) ERSPAN:

- VLAN sources are spanned only in the Rx direction.
- Session filtering functionality (VLAN or ACL filters) is supported only for Rx sources.
- A single forwarding engine instance supports four ERSPAN sessions. For Cisco Nexus 9300 Series switches, if the first three sessions have bidirectional sources, the fourth session has hardware resources only for Rx sources. This limitation might also apply to Cisco Nexus 9500 platform switches, depending on the ERSPAN source's forwarding engine instance mappings.
- An ERSPAN copy of Cisco Nexus 9300 platform switch 40G uplink interfaces will miss the dot1q information when spanned in the Rx direction.
- VLANs are supported as ERSPAN sources only in the ingress direction.

The following guidelines and limitations apply to FEX ports:

- If the sources used in bidirectional ERSPAN sessions are from the same FEX, the hardware resources are limited to two ERSPAN sessions.
- FEX ports are supported as ERSPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic.
- Cisco Nexus 9300 platform switches do not support ERSPAN destination being connected on a FEX interface. The ERSPAN destination must be connected to a front panel port.
- VLAN and ACL filters are not supported for FEX ports.

Priority flow control (PFC) ERSPAN has the following guidelines and limitations:

- PFC (Priority Flow Control) and LLFC (Link-Level Flow Control) are supported for all Cisco Nexus 9300 and 9500 platform switches except for the 100 Gb 9408PC line card and the 100 Gb M4PC generic expansion module (GEM).
- It is not supported on Cisco Nexus 9300 Series uplink ports.
- It cannot co-exist with filters.

- It is supported only in the Rx direction on physical or port-channel interfaces. It is not supported in the Rx direction on VLAN interfaces or in the Tx direction.

The following guidelines and limitations apply to Cisco Nexus 9200 Series switches:

- The **set-erspan-gre-proto** and **set-erspan-dscp** actions for ERSPAN ACLs are supported beginning with Cisco NX-OS Release 7.0(3)I4(1).
- UDF-based ERSPAN is supported beginning with Cisco NX-OS Release 7.0(3)I4(1).
- ERSPAN supports forward drops beginning with Cisco NX-OS Release 7.0(3)I4(1).
- Rx ERSPAN is not supported for multicast if the ERSPAN source and destination are on the same slice and no forwarding interface is on the slice. It is supported if a forwarding interface is on the slice or if the ERSPAN source and destination are on different slices.
- When multiple egress ports on the same slice are congested by egressing ERSPAN traffic, those egress ports will not get the line rate.
- The CPU ERSPAN source can be added only for the Rx direction (ERSPAN packets coming from the CPU).
- Using the ACL filter to span subinterface traffic on the parent interface is not supported.
- Multiple ACL filters are not supported on the same source.

The following guidelines and limitations apply to ERSPAN truncation:

- Truncation is supported only for Cisco Nexus 9300-EX and 9300-FX platform switches, beginning with Cisco NX-OS Release 7.0(3)I7(1).
- Truncation is supported only for local and ERSPAN source sessions. It is not supported for ERSPAN destination sessions.
- For ERSPAN sessions, the configured MTU value excludes the ERSPAN header. The egress packet for ERSPAN will have the MTU value + the number of bytes for the ERSPAN header.
- The bytes specified are retained starting from the header of the packets. The rest are truncated if the packet is longer than the MTU.
- The cyclic redundancy check (CRC) is recalculated for the truncated packet.

Default Settings

The following table lists the default settings for ERSPAN parameters.

Table 13: Default ERSPAN Parameters

Parameters	Default
ERSPAN sessions	Created in the shut state

Configuring ERSPAN

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Configuring an ERSPAN Source Session

You can configure an ERSPAN session on the local device only. By default, ERSPAN sessions are created in the shut state.

Note ERSPAN does not monitor any packets that are generated by the supervisor, regardless of their source.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**monitor erspan origin ip-address** *ip-address* **global** **Example:** `switch(config)# monitor erspan origin ip-address 10.0.0.1 global`	Configures the ERSPAN global origin IP address.
Step 3	**no monitor session** {*session-number* | **all**} **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified ERSPAN session. The new session configuration is added to the existing session configuration.

Step 4	**monitor session** {*session-number* | **all**} **type erspan-source** [**shut**]
	Example:
	`switch(config)# monitor session 3 type erspan-source`
	`switch(config-erspan-src)#`
		Configures an ERSPAN Type II source session. By default the session is bidirectional. The optional keyword shut specifies a shut state for the selected session.

Step 5	**description** *description* **Example:** `switch(config-erspan-src)# description erspan_src_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

	Command or Action	Purpose

Step 6	**source** {**interface** *type* [**tx** | **rx** |**both**] }
	Example:
	`switch(config-erspan-src)# source interface ethernet 2/1-3, ethernet 3/1 rx`
	Example:
	`switch(config-erspan-src)# source interface port-channel 2`
		You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress, egress, or both.
		For a unidirectional session, the direction of the source must match the direction specified in the session.

Step 7	(Optional) Repeat Step 7 to configure all ERSPAN sources.	—
Step 8	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1`	Configures the destination IP address in the ERSPAN session. Only one destination IP address is supported per ERSPAN source session.
Step 9	**erspan-id** *erspan-id* **Example:** `switch(config-erspan-src)# erspan-id 5`	Configures the ERSPAN ID for the ERSPAN source session. The ERSPAN range is from 1 to 1023.
Step 10	**vrf** *vrf-name* **Example:** `switch(config-erspan-src)# vrf default`	Configures the virtual routing and forwarding (VRF) instance that the ERSPAN source session uses for traffic forwarding. The VRF name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 11	(Optional) **ip ttl** *ttl-number* **Example:** `switch(config-erspan-src)# ip ttl 25`	Configures the IP time-to-live (TTL) value for the ERSPAN traffic. The range is from 1 to 255.
Step 12	(Optional) **ip dscp** *dscp-number* **Example:** `switch(config-erspan-src)# ip dscp 42`	Configures the differentiated services code point (DSCP) value of the packets in the ERSPAN traffic. The range is from 0 to 63.
Step 13	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN source session. By default, the session is created in the shut state.
Step 14	**exit** **Example:** `switch(config-erspan-src)# exit switch(config)#`	Exits the monitor configuration mode.
Step 15	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:**	Displays the ERSPAN session configuration.

	Command or Action	Purpose
	`switch(config)# show monitor session 3`	
Step 16	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the running ERSPAN configuration.
Step 17	(Optional) **show startup-config monitor** **Example:** `switch(config)# show startup-config monitor`	Displays the ERSPAN startup configuration.
Step 18	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Shutting Down or Activating an ERSPAN Session

You can shut down ERSPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, ERSPAN sessions are created in the shut state.

You can enable ERSPAN sessions to activate the copying of packets from sources to destinations. To enable an ERSPAN session that is already enabled but operationally down, you must first shut it down and then enable it. You can shut down and enable the ERSPAN session states with either a global or monitor configuration mode command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# monitor session 3 shut`	Shuts down the specified ERSPAN sessions. By default, sessions are created in the shut state.
Step 3	**no monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# no monitor session 3 shut`	Resumes (enables) the specified ERSPAN sessions. By default, sessions are created in the shut state. If a monitor session is enabled but its operational status is down, then to enable the session, you must first specify the **monitor**

	Command or Action	Purpose
		session shut command followed by the **no monitor session shut** command.

Step 4	**monitor session** *session-number* **type erspan-source**
	Example:
	`switch(config)# monitor session 3 type erspan-source`
	`switch(config-erspan-src)#`
		Enters the monitor configuration mode for the ERSPAN source type. The new session configuration is added to the existing session configuration.

Step 5	**shut** **Example:** `switch(config-erspan-src)# shut`	Shuts down the ERSPAN session. By default, the session is created in the shut state.
Step 6	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN session. By default, the session is created in the shut state.

Step 7	**exit**
	Example:
	`switch(config-erspan-src)# exit`
	`switch(config)#`
		Exits the monitor configuration mode.

Step 8	(Optional) **show monitor session all** **Example:** `switch(config)# show monitor session all`	Displays the status of ERSPAN sessions.
Step 9	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the ERSPAN running configuration.
Step 10	(Optional) **show startup-config monitor** **Example:** `switch(config)# show startup-config monitor`	Displays the ERSPAN startup configuration.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the ERSPAN Configuration

To display the ERSPAN configuration, perform one of the following tasks:

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the ERSPAN session configuration.
show running-config monitor	Displays the running ERSPAN configuration.
show startup-config monitor	Displays the ERSPAN startup configuration.

Configuration Examples for ERSPAN

Configuration Example for an ERSPAN Source Session Over IPv6

This example shows how to configure an ERSPAN source session over IPv6:

```
switch# configure terminal
switch(config)# monitor erspan origin ipv6-address 2001::10:0:0:9 global
switch(config)# moni session 10 type erspan-source
switch(config-erspan-src)# erspan-id 10
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# source interface ethernet 1/64
switch(config-erspan-src)# destination ip 9.1.1.2
```

Configuration Example for an ERSPAN ACL

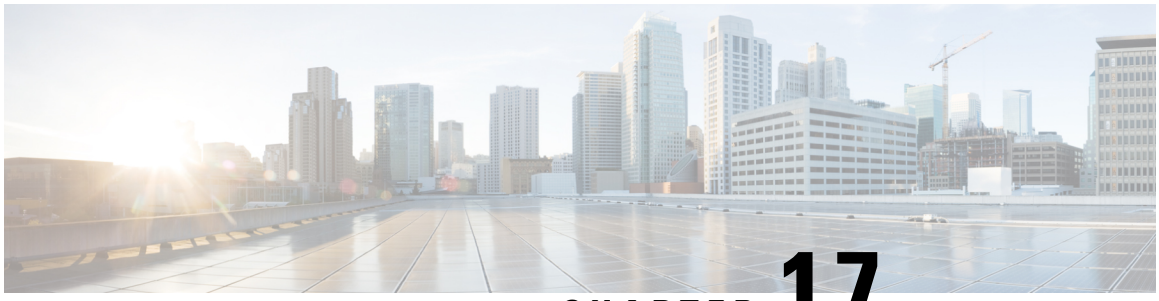
This example shows how to configure an ERSPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# vlan access-map erspan_filter 5
switch(config-access-map)# match ip address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# vlan access-map erspan_filter 10
switch(config-access-map)# match ip address match_12_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# filter access_group erspan_filter
```

Additional References

Related Documents

Related Topic	Document Title
FEX	*Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 9000 Series Switches*



CHAPTER 17

Configuring LLDP

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP) in order to discover other devices on the local network.

This chapter includes the following sections:

About LLDP

The Cisco Discovery Protocol (CDP) is a device discovery protocol that allows network management applications to automatically discover and learn about other Cisco devices that are connected to the network.

To permit the discovery of non-Cisco devices, the switch also supports the Link Layer Discovery Protocol (LLDP), a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard. LLDP allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP is a one-way protocol that transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

LLDP supports a set of attributes that it uses to discover other devices. These attributes contain type, length, and value (TLV) descriptions. LLDP devices can use TLVs to send and receive information to other devices on the network. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

LLDP advertises the following TLVs by default:

- DCBXP
- Management address
- Port description
- Port VLAN

- System capabilities
- System description
- System name

High Availability

The LLDP feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied.

For more information on high availability, see the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*.

Virtualization Support

One instance of LLDP is supported.

Guidelines and Limitations for LLDP

LLDP has the following configuration guidelines and limitations:

- LLDP must be enabled on the device before you can enable or disable it on any interfaces.
- LLDP is supported only on physical interfaces.
- LLDP can discover up to one device per port.
- DCBXP is not supported for LLDP.

Default Settings for LLDP

This table lists the LLDP default settings.

Parameters	Default
Global LLDP	Disabled
LLDP on interfaces	Enabled, after LLDP is enabled globally
LLDP hold time (before discarding)	120 seconds
LLDP reinitialization delay	2 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP TLVs	Enabled
LLDP receive	Enabled, after LLDP is enabled globally
LLDP transmit	Enabled, after LLDP is enabled globally
DCBXP	Enabled, provided LLDP is enabled

Parameters	Default
DCBXP version	Auto-detect

Configuring LLDP

Note Cisco NX-OS commands for this feature may differ from Cisco IOS commands for a similar feature.

Enabling or Disabling LLDP Globally

You can enable or disable LLDP globally on a device. You must enable LLDP globally to allow a device to send and receive LLDP packets.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature lldp** **Example:** `switch(config)# feature lldp`	Enables or disables LLDP on the device. LLDP is disabled by default.
Step 3	(Optional) **show running-config lldp** **Example:** `switch(config)# show running-config lldp`	Displays the global LLDP configuration. If LLDP is enabled, it shows "feature lldp." If LLDP is disabled, it shows an "Invalid command" error.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling LLDP on an Interface

After you globally enable LLDP, it is enabled on all supported interfaces by default. However, you can enable or disable LLDP on individual interfaces or selectively configure an interface to only send or only receive LLDP packets.

Before you begin

Make sure that you have globally enabled LLDP on the device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot/port*
	Example:
	`switch(config)# interface ethernet 7/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling LLDP and enters the interface configuration mode.

Step 3	[**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Enables or disables the transmission of LLDP packets on an interface. After you globally enable LLDP, it is enabled on all supported interfaces by default.
Step 4	[**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Enables or disables the reception of LLDP packets on an interface. After you globally enable LLDP, it is enabled on all supported interfaces by default.
Step 5	(Optional) **show lldp interface** *interface slot/port* **Example:** `switch(config-if)# show lldp interface ethernet 7/1`	Displays the LLDP configuration on the interface.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Optional LLDP Parameters

You can configure the frequency of LLDP updates, the amount of time for a receiving device to hold the information before discarding it, and the initialization delay time. You can also select the TLVs to include in LLDP packets.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) [**no**] **lldp holdtime** *seconds* **Example:** `switch(config)# lldp holdtime 200`	Specifies the amount of time in seconds that a receiving device should hold the information that is sent by your device before discarding it. The range is 10 to 255 seconds; the default is 120 seconds.
Step 3	(Optional) [**no**] **lldp reinit** *seconds* **Example:** `switch(config)# lldp reinit 5`	Specifies the delay time in seconds for LLDP to initialize on any interface. The range is 1 to 10 seconds; the default is 2 seconds.
Step 4	(Optional) [**no**] **lldp timer** *seconds* **Example:** `switch(config)# lldp timer 50`	Specifies the transmission frequency of LLDP updates in seconds. The range is 5 to 254 seconds; the default is 30 seconds.
Step 5	(Optional) **show lldp timers** **Example:** `switch(config)# show lldp timers`	Displays the LLDP hold time, delay time, and update frequency configuration.
Step 6	(Optional) [**no**] **lldp tlv-select** *tlv* **Example:** `switch(config)# lldp tlv-select system-name`	Specifies the TLVs to send and receive in LLDP packets. The available TLVs are management-address, port-description, port-vlan, system-capabilities, system-description, and system-name. All available TLVs are enabled by default.
Step 7	(Optional) **show lldp tlv-select** **Example:** `switch(config)# show lldp tlv-select`	Displays the LLDP TLV configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the LLDP Configuration

To display the LLDP configuration, perform one of the following tasks:

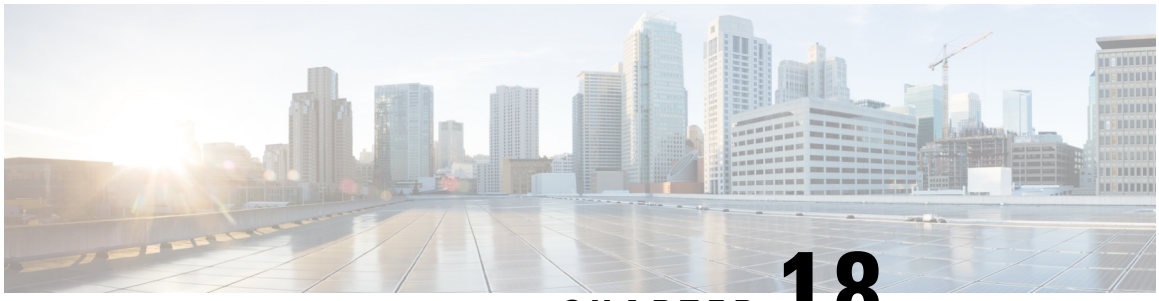
Command	Purpose
show running-config lldp	Displays the global LLDP configuration.
show lldp interface *interface slot/port*	Displays the LLDP interface configuration.
show lldp timers	Displays the LLDP hold time, delay time, and update frequency configuration.
show lldp tlv-select	Displays the LLDP TLV configuration.
show lldp neighbors {**detail** | **interface** *interface slot/port*}	Displays the LLDP neighbor device status.
show lldp traffic	Displays the LLDP counters, including the number of LLDP packets sent and received by the device, the number of discarded packets, and the number of unrecognized TLVs.
show lldp traffic interface *interface slot/port*	Displays the number of LLDP packets sent and received on the interface.

Use the **clear lldp counters** command to clear the LLDP statistics.

Configuration Example for LLDP

This example shows how to enable LLDP on a device; disable LLDP on some interfaces; configure optional parameters such as hold time, delay time, and update frequency; and disable several LLDP TLVs:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature lldp
switch(config)# interface ethernet 7/9
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# interface ethernet 7/10
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# lldp holdtime 200
switch(config)# lldp reinit 5
switch(config)# lldp timer 50
switch(config)# no lldp tlv-select port-vlan
switch(config)# no lldp tlv-select system-name
```



CHAPTER 18

Performing Software Maintenance Upgrades

This chapter describes how to perform software maintenance upgrades (SMUs) on Cisco NX-OS devices.

This chapter includes the following sections:

About SMUs

A software maintenance upgrade (SMU) is a package file that contains fixes for a specific defect. SMUs are created to respond to immediate issues and do not include new features. Typically, SMUs do not have a large impact on device operations. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.

The effect of an SMU depends on its type:

- Process restart SMU-Causes a process or group of processes to restart on activation.
- Reload SMU-Causes a parallel reload of supervisors and line cards.

SMUs are not an alternative to maintenance releases. They provide a quick resolution of critical issues. All defects fixed by SMUs are integrated into the next maintenance releases of upcoming software trains, as applicable. SMUs also have the following considerations:

- SMUs are created for the following:
  - Critical SIR PSIRTs without a workaround or fix
  - Severity1 and Severity2 issues without a workaround or fix
- If a fix is already available in a maintenance release of the same software train or already released on a later long-lived release, no SMU is provided. You are encouraged to acquire the fix from the maintenance release.

Note Depending on the fix, in some cases it may not be possible to provide an SMU. In such cases, the only option is to upgrade to the next maintenance release when available.

For information on upgrading your device to a new feature or maintenance release, see the Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide.

Note Activating an SMU does not cause any earlier SMUs, or the package to which the SMU applies, to be automatically deactivated.

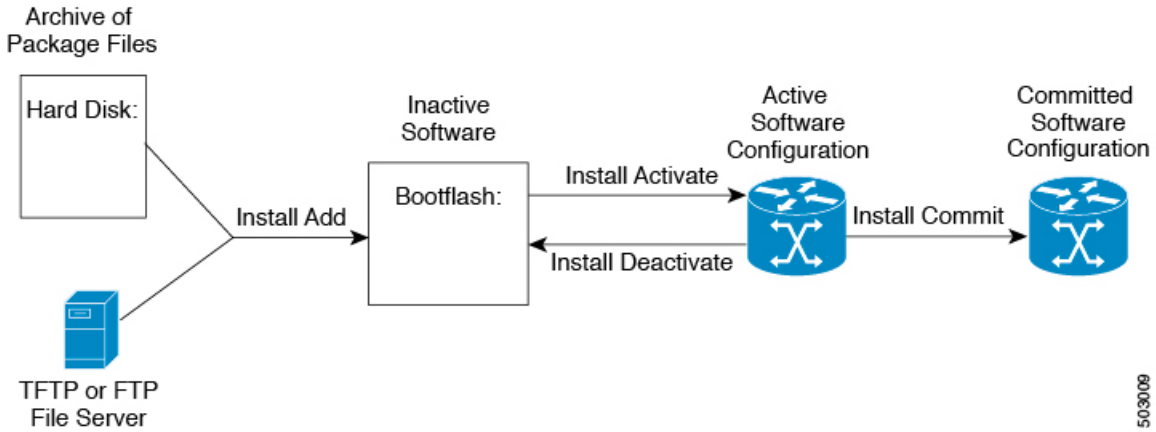
Package Management

The general procedure for adding and activating SMU packages on the device is as follows:

1. Copy the package file or files to a local storage device or file server.
2. Add the package or packages on the device using the **install add** command.
3. Activate the package or packages on the device using the **install activate** command.
4. Commit the current set of packages using the **install commit** command.
5. (Optional) Deactivate and remove the package.

The following figure illustrates the key steps in the package management process.

Figure 5: Process to Add, Activate, and Commit SMU Packages



Impact of Package Activation and Deactivation

The activation or deactivation of an SMU package can have an immediate impact on the system. The system can be affected in the following ways:

- New processes might be started.
- Running processes might be stopped or restarted.
- All processes in the line cards might be restarted. Restarting processes in the line cards is equivalent to a soft reset.
- The line cards might reload.
- No processes in the line cards might be affected.

Note You must address any issues that result from the revised configuration and reapply the configuration, if necessary.

Tip After the activation process completes, enter the **show install log** command to display the process results.

Prerequisites for SMUs

These prerequisites must be met for a package to be activated or deactivated:

- You must be in a user group associated with a task group that includes the proper task IDs. If you suspect a user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Verify that all line cards are installed and operating properly. For example, do not activate or deactivate packages while line cards are booting, while line cards are being upgraded or replaced, or when you anticipate an automatic switchover activity.

Guidelines and Limitations for SMUs

SMUs have the following guidelines and limitations:

- Some packages require the activation or deactivation of other packages. If the SMUs have dependencies on each other, you cannot activate them without first activating the previous ones.
- The package being activated must be compatible with the current active software set.
- Activation is performed only after the package compatibility checks have been passed. If a conflict is found, an error message displays.
- While a software package is being activated, other requests are not allowed to run on any of the impacted nodes. Package activation is completed when a message similar to this one appears:

```
Install operation 1 completed successfully at Thu Jan 9 01:19:24 2014
```

- Each CLI install request is assigned a request ID, which can be used later to review the events.

- If you perform a software maintenance upgrade and later upgrade your device to a new Cisco NX-OS software release, the new image will overwrite both the previous Cisco NX-OS release and the SMU package file.
- For the "Unable to remove MAC ACE using sequence number in 7.0(3)I7(2)" issue, if you are going to apply the patch that resolves it, you must make sure that the ACL is deleted before applying the patch. Otherwise, the issue will be seen again. This issue applies only to the ACL which has the redirect keyword in it.

Performing a Software Maintenance Upgrade for Cisco NX-OS

Preparing for Package Installation

You should use several **show** commands to gather information in preparation for the SMU package installation.

Before you begin

Determine if a software change is required.

Verify that the new package is supported on your system. Some software packages require that other packages or package versions be activated, and some packages support only specific line cards.

Review the release notes for important information related to that release and to help determine the package compatibility with your device configuration.

Verify that the system is up, stable, and prepared for the software changes.

Procedure

	Command or Action	Purpose
Step 1	**show logging logfile | grep -i "System ready"** **Example:** `switch# show logging logfile | grep -i "System ready"`	Displays if your system is up. Use this command to verify that the system is ready for SMU package installation. Configuring install commands before the system is ready, may result with an "Install operation 11 failed because cannot lock config" error message.
Step 2	**show install active** **Example:** `switch# show install active`	Displays the active software on the device. Use this command to determine what software should be added on the device and to compare to the active software report after installation operations are complete.
Step 3	**show module** **Example:** `switch# show module`	Confirms that all modules are in the stable state.

	Command or Action	Purpose
Step 4	**show clock** **Example:** `switch# show clock`	Verifies that the system clock is correct. Software operations use certificates based on device clock times.

Example

This example shows how to verify that the system is up. A "System ready" response indicates that the system is ready for SMU package installation.

```
switch# show logging logfile | grep -i "System ready"
2018 Feb 19 11:13:04 switch %ASCII-CFG-2-CONF_CONTROL: System ready
```

This example shows how to display the active packages for the entire system. Use this information to determine if a software change is required.

```
switch# show install active
Boot Image:
        NXOS Image: bootflash:///nxos.7.0.3.I7.3.1.bin

Active Packages:

switch#
```

This example shows how to display the current system clock setting:

```
switch# show clock
02:14:51.474 PST Wed Jan 04 2014
```

Downloading the SMU Package File from Cisco.com

Follow these steps to download the SMU package file:

Procedure

Step 1 Log in to Cisco.com.

Step 2 Go to the Download Software page at this URL: http://software.cisco.com/download/navigator.html

Step 3 In the Select a Product list, choose **Switches** > **Data Center Switches** > **Cisco Nexus 9000 Series Switches** > *model*.

Step 4 Choose the appropriate SMU file for your device and click **Download**.

Copying the Package File to a Local Storage Device or Network Server

You must copy the SMU package file to a local storage device or a network file server to which the device has access. After this task is done, the package can be added and activated on the device.

If you need to store package files on the device, we recommend that you store the files on the hard disk. The boot device is the local disk from which the package is added and activated. The default boot device is bootflash:.

Tip Before you copy package files to a local storage device, use the **dir** command to determine if the required package files are already on the device.

If the SMU package files are located on a remote TFTP, FTP, or SFTP server, you can copy the files to a local storage device. After the files are located on the local storage device, the package can be added and activated on the device from that storage device. The following server protocols are supported:

- Trivial File Transfer Protocol—TFTP allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password). It is a simplified version of FTP.

Note Some package files might be larger than 32 MB, and the TFTP services provided by some vendors might not support a file this large. If you do not have access to a TFTP server that supports files larger than 32 MB, download the file using FTP.

- File Transfer Protocol—FTP is part of the TCP/IP protocol stack and requires a username and password.
- SSH File Transfer Protocol—SFTP is part of the SSHv2 feature in the security package and provides for secure file transfers. For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Note Consult your system administrator for the location and availability of your network server.

Use the commands in the following table to copy the SMU package file from the server to your device using the file transfer protocols.

Table 14: Commands for Copying SMU Package Files to the Device

Command	Purpose
copy tftp://*hostname-or-ipaddress*/*directory-path*/*filename* **bootflash:** `switch# copy tftp://10.1.1.1/images/ n9000-dk9.6.1.2.I2.1.CSCab00001.bin bootflash:`	Copies the package file from the TFTP server to the bootflash:. • *hostname-or-ipaddress*—The hostname or IP address of the network file server. • *directory-path*—The network file server path that leads to the package file to be added. • *filename*—The name of the package file that you want to add.

Command	Purpose
copy ftp://*username*:*password*@*hostname-or-ipaddress*/*directory-path*/*filename* **bootflash:** `switch# copy ftp://john:secret@10.1.1.1/images/ n9000-dk9.6.1.2.I2.1.CSCab00001.bin bootflash:`	Copies the package file from the FTP server to the bootflash:. • *username*—The username of the user who has access privileges to the directory in which the package file is stored. • *password*—The password associated with the username of the user who has access privileges to the directory in which the package file is stored. If a password is not provided, the networking device accepts anonymous FTP. • *hostname-or-ipaddress*—The hostname or IP address of the network file server. • *directory-path*—The network file server path that leads to the package file to be added. The specified directory should be a directory under the home directory of the user. In this example, the file being downloaded is in a subdirectory called "images" in the home directory of the user "john." **Note** For FTP services, *directory-path* is the directory relative to the *username* home directory. If you want to specify an absolute path for the directory, you must add a “/” following the server address. • *filename*—The name of the package file that you want to add.

Command	Purpose
copy sftp://*hostname-or-ipaddress*/*directory-path*/*filename* **bootflash:** `switch# copy sftp://10.1.1.1/images/n9000-dk9.6.1.2.I2.1.CSCab00001.bin bootflash:`	Copies the package file from the SFTP server to the bootflash:. • *hostname-or-ipaddress*—The hostname or IP address of the network file server. • *directory-path*—The network file server path that leads to the package file to be added. • *filename*—The name of the package file that you want to add.

After the SMU package file has been transferred to a network file server or the local storage device, you are ready to add and activate the file.

Adding and Activating Packages

You can add SMU package files that are stored on a local storage device or on a remote TFTP, FTP, or SFTP server to your device.

Note The SMU package being activated must be compatible with the currently active software to operate. When an activation is attempted, the system runs an automatic compatibility check to ensure that the package is compatible with the other active software on the device. If a conflict is found, an error message displays. The activation is performed only after all compatibility checks have been passed.

Note Activating an SMU does not cause any earlier SMUs or the package to which the SMU applies to be automatically deactivated.

Before you begin

Make sure that all packages to be added are present on a local storage device or a network file server.

Make sure that you meet all of the prerequisites for the activation of packages.

Complete the procedure described in Copying the Package File to a Local Storage Device or Network Server, on page 211.

Procedure

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session to the console port.

	Command or Action	Purpose
Step 2	(Optional) **dir bootflash:**	Displays the package files that are available to be added. **Note** Only SMU package files can be added and activated using this procedure.
Step 3	**install add** *filename* [**activate**] **Example:** `switch# install add bootflash: n9000-dk9.6.1.2.I2.1.CSCab00001.bin`	Unpacks the package software files from the local storage device or network server and adds them to the bootflash: and all active and standby supervisors installed on the device. The *filename* argument can take any of these formats: • **bootflash:***filename* • **tftp://***hostname-or-ipaddress/directory-path/filename* • **ftp://***username***:***password@ hostname-or-ipaddress/directory-path/filename* • **usb1:***filename* • **usb2:***filename* For all SMU packages except the CSCur02700 SMU package, you can use the optional **activate** keyword to automatically activate the package after it is added successfully. **Note** For the CSCur02700 SMU package, use the **install activate** command in Step 5 to activate the package. Do not use the optional **activate** keyword with the **install add** command as the package might fail and require a reboot. Multiple versions of an SMU package can be added to the storage device without impacting the running configuration, but only one version of a package can be activated for a line card. **Note** Press **?** after a partial package name to display all possible matches available for activation. If there is only one match, press the **Tab** key to fill in the rest of the package name.
Step 4	(Optional) **show install inactive** **Example:** `switch# show install inactive`	Displays the inactive packages on the device. Verify that the package added in the previous step appears in the display.

	Command or Action	Purpose

Step 5	Required: **install activate** *filename*
	Example:
	`switch# install activate n9000-dk9.6.1.2.I2.1.CSCab00001.bin`
	Example:
	`switch# install activate n9000-dk9.6.1.2.I2.1.CSCab00001.bin Install operation 18 !!WARNING!! This patch will get activated only after a reload of the switch. at Sun Mar 9 00:42:12 2014`
		Activates a package that was added to the device. SMU packages remain inactive until activated. (Skip this step if the package was activated earlier with the **install add activate** command.)
		Tip After the activation process finishes, enter the **show install log** command to display the process results.

Step 6	Repeat Step 5 until all packages are activated.	Activates additional packages as required.
Step 7	(Optional) **show install active** **Example:** `switch# show install active`	Displays all active packages. Use this command to determine if the correct packages are active.

Committing the Active Package Set

When an SMU package is activated on the device, it becomes part of the current running configuration. To make the package activation persistent across system-wide reloads, you must commit the package on the device.

Note On startup, the device loads the committed package set. If the system is reloaded before the current active package is committed, the previously committed package set is used.

Before you begin

Before you commit a package set, verify that the device is operating correctly and is forwarding packets as expected.

Complete the procedure described in Adding and Activating Packages, on page 214.

Procedure

	Command or Action	Purpose
Step 1	**install commit** *filename* **Example:** `switch# install commit n9000-dk9.6.1.2.I2.1.CSCab00001.bin`	Commits the current set of packages so that these packages are used if the device is restarted.
Step 2	Required: **reload module** *standby-sup-slot* **Example:**	Reloads the standby supervisor module, if installed.

	Command or Action	Purpose
	`switch# reload module 2`	**Note** If you are applying the SMU package in a dual-supervisor system and your device is running Cisco NX-OS Release 6.1(2)I2(3) or a software release prior to Cisco NX-OS 6.1(2)I2(2b), you must reload the standby supervisor module.
Step 3	(Optional) **show install committed** **Example:** `switch# show install committed`	Displays which packages are committed.

Example

This example shows how to commit active SMU packages on the device and then verify the committed packages:

```
switch# install commit n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 2 completed successfully at Thu Jan 9 01:20:46 2014

switch# show install committed
Committed Packages:
n9000-dk9.6.1.2.I2.1.CSCab00001.bin
```

Deactivating and Removing Packages

When a package is deactivated, it is no longer active on the device, but the package files remain on the boot disk. The package files can be reactivated later, or they can be removed from the disk.

The Cisco NX-OS software also provides the flexibility to roll back the selected package set to a previously saved package set. If you find that you prefer a previous package set over the currently active package set, you can use the **install deactivate** and **install commit** commands to make a previously active package set active again.

Before you begin

You cannot deactivate a package if it is required by another active package. When you attempt to deactivate a package, the system runs an automatic check to ensure that the package is not required by other active packages. The deactivation is performed only after all compatibility checks have been passed.

You cannot delete a package if it is part of the running or committed software of the device.

Procedure

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session to the console port.

	Command or Action	Purpose
Step 2	Required: **reload module** *standby-sup-slot* **Example:** `switch# reload module 2`	Reloads the standby supervisor module, if installed. **Note** If you are deactivating the SMU package in a dual-supervisor system and your device is running Cisco NX-OS Release 6.1(2)I2(3) or a software release prior to Cisco NX-OS 6.1(2)I2(2b), you must reload the standby supervisor module.
Step 3	**install deactivate** *filename* **Example:** `switch# install deactivate n9000-dk9.6.1.2.I2.1.CSCab00001.bin`	Deactivates a package that was added to the device and turns off the package features for the line card. **Note** You must run **install commit** after **install deactivate** to deactivate the package completely, otherwise the package gets activated again after reload. For reload SMU, run **install commit** after the device reloads.
Step 4	(Optional) **show install inactive** **Example:** `switch# show install inactive`	Displays the inactive packages on the device.
Step 5	(Optional) **install commit** **Example:** `switch# install commit`	Commits the current set of packages so that these packages are used if the device is restarted. **Note** Packages can be removed only if the deactivation operation is committed.

Step 6	(Optional) **install remove** {*filename* | **inactive**}
	Example:
	`switch# install remove n9000-dk9.6.1.2.I2.1.CSCab00001.bin`
	`Proceed with removing n9000-dk9.6.1.2.I2.1.CSCab00001.bin? (y/n)? [n] y`
	Example:
	`switch# install remove inactive`
	`Proceed with removing? (y/n)? [n] y`
		Removes the inactive package.
		• Only inactive packages can be removed.
		• Packages can be removed only if they are deactivated from all line cards in the device.
		• The package deactivation must be committed.
		• To remove a specific inactive package from a storage device, use the **install remove** command with the *filename* argument.
		• To remove all inactive packages from all nodes in the system, use the **install**

	Command or Action	Purpose
		remove command with the **inactive** keyword.

Displaying Installation Log Information

The installation log provides information on the history of the installation operations. Each time an installation operation is run, a number is assigned to that operation.

- Use the **show install log** command to display information about both successful and failed installation operations.
- Use the **show install log** command with no arguments to display a summary of all installation operations. Specify the *request-id* argument to display information specific to an operation. Use the **detail** keyword to display details for a specific operation, including file changes, nodes that could not be reloaded, and any impact to processes.

This example shows how to display information for all installation requests:

```
switch# show install log
Thu Jan 9 01:26:09 2014
Install operation 1 by user 'admin' at Thu Jan 9 01:19:19 2014
Install add bootflash:n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 1 completed successfully at Thu Jan 9 01:19:24 2014
----------------------------------------
Install operation 2 by user 'admin' at Thu Jan 9 01:19:29 2014
Install activate n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 2 completed successfully at Thu Jan 9 01:19:45 2014
----------------------------------------
Install operation 3 by user 'admin' at Thu Jan 9 01:20:05 2014
Install commit n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 3 completed successfully at Thu Jan 9 01:20:08 2014
----------------------------------------
Install operation 4 by user 'admin' at Thu Jan 9 01:20:21 2014
Install deactivate n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 4 completed successfully at Thu Jan 9 01:20:36 2014
----------------------------------------
Install operation 5 by user 'admin' at Thu Jan 9 01:20:43 2014
Install commit n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 5 completed successfully at Thu Jan 9 01:20:46 2014
----------------------------------------
Install operation 6 by user 'admin' at Thu Jan 9 01:20:55 2014
Install remove n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 6 completed successfully at Thu Jan 9 01:20:57 2014
----------------------------------------
Install operation 7 by user 'admin' at Thu Jan 9 01:21:07 2014
Install remove
Install operation 7 completed successfully at Thu Jan 9 01:21:10 2014
```

This example shows how to display additional information, including any impact to nodes and processes:

```
switch# show install log detail
Thu Jan 9 01:24:03 2014
Install operation 1 by user 'admin' at Thu Jan 9 01:19:19 2014
Installer started downloading the package: /n9000-dk9.6.1.2.I2.1.CSCab00001.bin
via bootflash
Install add bootflash:n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Copying file at Thu Jan 9 01:19:20 2014
```

```
Download success, 238545 bytes received
Verifying package
Checking MD5 at Thu Jan 9 01:19:21 2014
MD5 checksum OK
Checking HW platform at Thu Jan 9 01:19:22 2014
Checking SW platform at Thu Jan 9 01:19:23 2014
Package verified successfully
Sending patch file to plugin manager at Thu Jan 9 01:19:23 2014
The following package is now available to be activated: n9000-dk9.6.1.2.I2.1.CSC
ab00001.bin
Install operation 1 completed successfully at Thu Jan 9 01:19:24 2014
----------------------------------------
Install operation 2 by user 'admin' at Thu Jan 9 01:19:29 2014
Install activate n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install activate action started
The software will be activated with process restart
2 processes affected
sysinfo (modified)
vman (modified)
Install operation 2 completed successfully at Thu Jan 9 01:19:45 2014
----------------------------------------
Install operation 3 by user 'admin' at Thu Jan 9 01:20:05 2014
Install commit n9000-dk9.6.1.2.I2.1.CSCab00001.bin
MD5 checksum OK for patch: n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 3 completed successfully at Thu Jan 9 01:20:08 2014
----------------------------------------
Install operation 4 by user 'admin' at Thu Jan 9 01:20:21 2014
Install deactivate n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install deactivate action started
The software will be deactivated with process restart
2 processes affected
sysinfo (modified)
vman (modified)
Install operation 4 completed successfully at Thu Jan 9 01:20:36 2014
----------------------------------------
Install operation 5 by user 'admin' at Thu Jan 9 01:20:43 2014
Install commit n9000-dk9.6.1.2.I2.1.CSCab00001.bin
MD5 checksum OK for patch: n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 5 completed successfully at Thu Jan 9 01:20:46 2014
----------------------------------------
Install operation 6 by user 'admin' at Thu Jan 9 01:20:55 2014
Install remove n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install operation 6 completed successfully at Thu Jan 9 01:20:57 2014
----------------------------------------
Install operation 7 by user 'admin' at Thu Jan 9 01:21:07 2014
Install remove
Install operation 7 completed successfully at Thu Jan 9 01:21:10 2014
```

This example shows the output after an SMU package has been activated but before the switch has been reloaded:

```
switch# show install log detail
Install operation 18 by user 'admin' at Sun Mar 9 00:42:10 2014
Install activate n9000-dk9.6.1.2.I2.1.CSCab00001.bin
Install activate action started
The software will be activated with system reload
Install operation 18 !!WARNING!! This patch will get activated only after
a reload of the switch. at Sun Mar 9 00:42:12 2014
```

Performing a Software Maintenance Upgrade for Guest Shell Bash

You can perform a software maintenance upgrade for Bash in the Guest Shell.

Procedure

	Command or Action	Purpose
Step 1	Download the SMU package file for Guest Shell Bash from Cisco.com.	Obtains the package file from Cisco.com. For instructions, see Downloading the SMU Package File from Cisco.com, on page 211.
Step 2	Copy the SMU package file to the bootflash: of the switch.	Copies the package file to the device. For instructions, see Copying the Package File to a Local Storage Device or Network Server, on page 211.

Step 3	**guestshell**
	Example:
	`switch# guestshell`
	`guestshell:~$`
		Accesses the Guest Shell.

Step 4	**sudo rpm -Uvh** /bootflash/*filename*
	Example:
	`guestshell:~$ sudo rpm -Uvh /bootflash/bash-4.2-r8.x86_64.rpm`
	`Preparing...`
	`########################################## [100%]`
	`1:bash`
	`########################################## [100%]`
	`update-alternatives: Linking //bin/sh to /bin/bash`
		Upgrades the existing Bash file in the Guest Shell.

Step 5	**rpm -qa | grep bash**
	Example:
	`guestshell:~$ rpm -qa | grep bash`
	`bash-4.2-r8.x86_64`
		Verifies that the new version of the Bash file was installed successfully.

Step 6	**guestshell sync**
	Example:
	`switch# guestshell sync`
	`Access to the guest shell will be temporarily disabled while it synchronizes contents to standby.`
	`Are you sure you want to continue? (y/n)`
		On a dual-supervisor system, synchronizes the rootfs with the Bash SMU version to the standby supervisor before doing a switchover. If you do not run this command, you will need to repeat this procedure after a supervisor switchover.

	Command or Action	Purpose
	[n] y dt-n9k3-1# 2014 Oct 7 05:00:01 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-INSTALL_STATE: Deactivating virtual service 'guestshell+' dt-n9k3-1# 2014 Oct 7 05:00:06 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully deactivated virtual service 'guestshell+' 2014 Oct 7 05:00:12 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully deactivated virtual service 'guestshell+' ; Starting sync to standby sup 2014 Oct 7 05:00:32 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-MOVE_STATE: Successfully synced virtual service 'guestshell+' ; Activating 2014 Oct 7 05:00:32 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Activating virtual service 'guestshell+' 2014 Oct 7 05:00:56 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully activated virtual service 'guestshell+'	**Note** The new Bash file is preserved after a Guest Shell reboot or Guest Shell disable+enable. However, you need to reinstall the Guest Shell Bash SMU package file after a Guest Shell destroy+enable.

Additional References

Related Documents

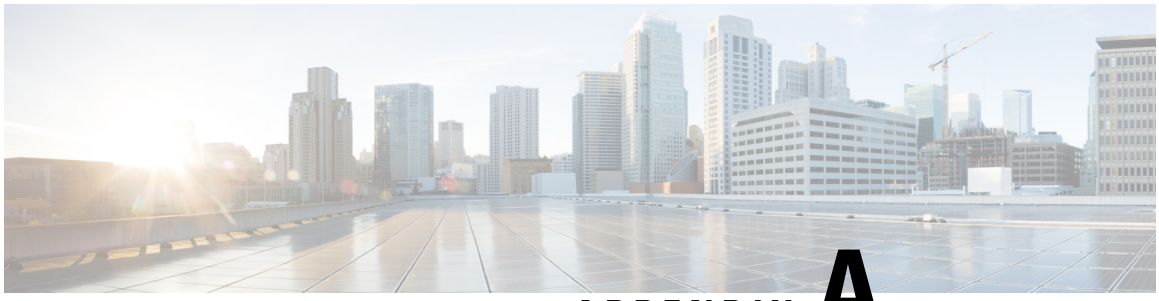
Related Topic	Document Title
Software upgrades	*Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*

SMU History

This table lists the release history for SMU package files.

SMU Package File	Releases	Description
bash-4.2-r8.x86_64.rpm	6.1(2)I3(1)	Guest Shell Bash SMU for Bash vulnerabilities CVE-2014-6277, CVE-2014-6278, CVE-2014-7186, and CVE-2014-7187
n9000-dk9.6.1.2.I3.1.CSCur02700.bin	6.1(2)I3(1) and all 6.1(2)I2(x) releases	Cisco NX-OS SMU for CSCur02700 (Bash vulnerabilities CVE-2014-6277, CVE-2014-6278, CVE-2014-7186, and CVE-2014-7187)

SMU Package File	Releases	Description
n9000-dk9.6.1.2.I2.1.CSCup81353.bin	6.1(2)I2(1), 6.1(2)I2(2), 6.1(2)I2(2a), and 6.1(2)I2(3)	Cisco NX-OS SMU for CSCup81353



APPENDIX A

IETF RFCs supported by Cisco NX-OS System Management

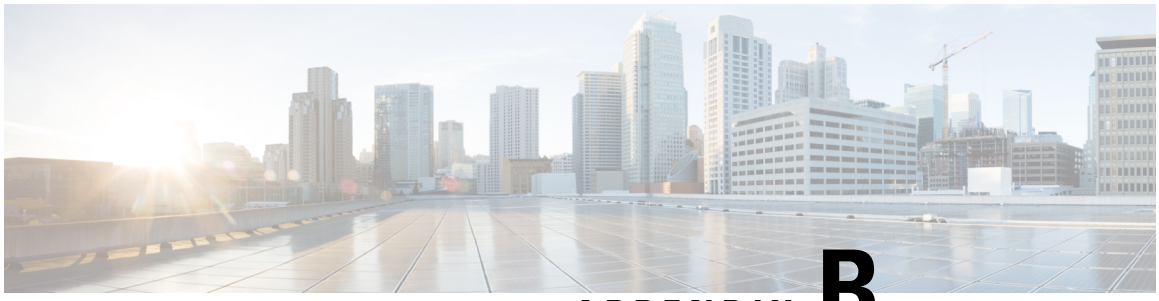
This appendix lists the IETF RFCs for system management supported in Cisco NX-OS.

IETF RFCs Supported by Cisco NX-OS System Management

This appendix lists the IETF RFCs for system management supported in Cisco NX-OS.

RFCs	Title
RFC 2819	*Remote Network Monitoring Management Information Ba*
RFC 3411 and RFC 3418	*An Architecture for Describing Simple Network Managem (SNMP) Management Frameworks*

IETF RFCs supported by Cisco NX-OS System Management



APPENDIX B

Embedded Event Manager System Events and Configuration Examples

This appendix describes the Embedded Event Manager (EEM) system policies, events, and policy configuration examples.

This appendix includes the following sections:

EEM System Policies

The following table lists the Embedded Event Manager (EEM) system policies.

Event	Description
__PortLoopback	Do CallHome, log error in Syslog/OBFL/Exception Log, and disable further HM testing on affected ports after 10 consecutive failures of GOLD "PortLoopback" test
__RewriteEngineLoopback	Do CallHome, log error in Syslog/OBFL/Exception Log, and disable further HM testing on affected ports after 10 consecutive failures of GOLD "RewriteEngine" test
__asic_register_check	Do CallHome, log error, and disable further HM testing for that ASIC device/instance after 20 consecutive failures of GOLD "ASICRegisterCheck" test
__compact_flash	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "CompactFlash" test
__crypto_device	Do CallHome and log error when GOLD "CryptoDevice" test fails

Embedded Event Manager System Events and Configuration Examples

Event	Description
__eobc_port_loopback	Do CallHome and log error when GOLD "EOBCPortLoopback" test fails
__ethpm_debug_1	Action: none
__ethpm_debug_2	Action: none
__ethpm_debug_3	Action: none
__ethpm_debug_4	Action: none
__ethpm_link_flap	More than 30 link flaps in a 420-second interval. Action: Error. Disable the port
__external_compact_flash	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "ExternalCompactFlash" test
__lcm_module_failure	Power cycle two times and then power down
__management_port_loopback	Do CallHome and log error when GOLD "ManagementPortLoopback" test fails
__nvram	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "NVRAM" test
__pfm_fanabsent_all_systemfan	Shuts down if both fan trays (f1 and f2) are absent for 2 minutes
__pfm_fanbad_all_systemfan	Syslog when fan goes bad
__pfm_fanbad_any_singlefan	Syslog when fan goes bad
__pfm_power_over_budget	Syslog warning for insufficient power overbudget
__pfm_tempev_major	TempSensor Major Threshold. Action: Shutdown
__pfm_tempev_minor	TempSensor Minor Threshold. Action: Syslog
__primary_bootrom	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "PrimaryBootROM" test
__pwr_mgmt_bus	Do CallHome, log error, and disable further HM testing for the module or spine-card after 20 consecutive failures of GOLD "PwrMgmtBus" test
__real_time_clock	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "RealTimeClock" test

Event	Description
__secondary_bootrom	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "SecondaryBootROM" test
__spine_control_bus	Do CallHome, log error, and disable further HM testing for that module or spine-card after 20 consecutive failures of GOLD "SpineControlBus" test
__standby_fabric_loopback	Do CallHome, log error, and disable further HM testing after 10 consecutive failures
__status_bus	Do CallHome, log error, and disable further HM testing after 5 consecutive failures of GOLD "StatusBus" test
__system_mgmt_bus	Do Call Home, log error, and disable further HM testing for that fan or power supply after 20 consecutive failures of GOLD "SystemMgmtBus" test
__usb	Do Call Home and log error when GOLD "USB" test fails

EEM Events

The following table describes the EEM events you can use on the device.

EEM Event	Description
application	Publishes an application-specific event.
cli	CLI command is entered that matches a pattern with a wildcard.
counter	EEM counter reaches a specified value or range.
fanabsent	System fan tray is absent.
fanbad	System fan generates a fault.
fib	Monitors routes or TCAM usage in the unicast FIB.
gold	GOLD test failure condition is hit.
interface	Interface counter exceeds a threshold.
memory	Available system memory exceeds a threshold.
module	Specified module enters the selected status.
module-failure	Module failure is generated.

EEM Event	Description
none	Runs the policy event without any events specified.
oir	Online insertion or removal occurs.
policy-default	Default parameters and thresholds are used for the events in the system policy you override.
poweroverbudget	Platform software detects a power budget condition.
snmp	SNMP object ID (OID) state changes.
storm-control	Platform software detects an Ethernet packet storm condition.
syslog	Monitors syslog messages and invokes the policy based on the search string in the policy.
sysmgr	System manager generates an event.
temperature	Temperature level in the system exceeds a threshold.
timer	Specified time is reached.
track	Tracked object changes state.

Configuration Examples for EEM Policies

Configuration Examples for CLI Events

Monitoring Interface Shutdown

This example shows how to monitor an interface shutdown:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# event manager applet monitorShutdown
switch(config-applet)#
switch(config-applet)# description "Monitors interface shutdown."
switch(config-applet)# event cli match "conf t; interface *; shutdown"
switch(config-applet)# action 1.0 cli show interface e 3/1
switch(config)# copy running-config startup-config
```

Note Outputs of **show** commands entered as part of EEM policy are archived in the logflash as text files with the "eem_archive_" prefix. To view the archived output, use the **show file logflash:eem_archive_***n* command.

Monitoring Module Powerdown

This example shows how to monitor a module powerdown:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# event manager applet monitorPoweroff
switch(config-applet)#
switch(config-applet)# description "Monitors module power down."
switch(config-applet)# event cli match "conf t; poweroff *"
switch(config-applet)# action 1.0 cli show module
switch(config)# copy running-config startup-config
```

Adding a Trigger to Initiate a Rollback

This example shows how to add a trigger to initiate a rollback:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)#
switch(config)# event manager applet rollbackTrigger
switch(config-applet)#
switch(config-applet)# description "Rollback trigger."
switch(config-applet)# event cli match "rollback *"
switch(config-applet)# action 1.0 cli copy running-config bootflash:last_config
switch(config)# copy running-config startup-config
```

Configuration Examples to Override (Disable) Major Thresholds

Preventing a Shutdown When Reaching a Major Threshold

This example shows how to prevent a shutdown caused by reaching a major threshold:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Disabling One Bad Sensor

This example shows how to disable only sensor 3 on module 2 when sensor 3 is malfunctioning (all other sensors are unaffected):

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
```

```
switch(config)# end
```

Disabling Multiple Bad Sensors

This example shows how to disable sensors 5, 6, and 7 on module 2 when these sensors are malfunctioning (all other sensors are unaffected):

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 5 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 6 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Overriding (Disabling) an Entire Module

This example shows how to disable module 2 when it is malfunctioning:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Overriding (Disabling) Multiple Modules and Sensors

This example shows how to disable sensors 3, 4, and 7 on module 2 and all sensors on module 3 when they are malfunctioning:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 4 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
```

```
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 3 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Enabling One Sensor While Disabling All Remaining Sensors of All Modules

This example shows how to disable all sensors on all modules except sensor 4 on module 9:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 4 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Enabling Multiple Sensors While Disabling All Remaining Sensors of All Modules

This example shows how to disable all sensors on all modules except sensors 4, 6, and 7 on module 9:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 4 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 6 threshold major
switch(config-applet)# action 3 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet4 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 7 threshold major
switch(config-applet)# action 4 policy-default
switch(config-applet)# end
```

Enabling All Sensors of One Module While Disabling All Sensors of the Remaining Modules

This example shows how to disable all sensors on all modules except all sensors on module 9:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 threshold major
```

```
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Enabling a Combination of Sensors on Modules While Disabling All Sensors of the Remaining Modules

This example shows how to disable all sensors on all modules except sensors 3, 4, and 7 on module 2 and all sensors on module 3:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 4 threshold major
switch(config-applet)# action 3 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet4 override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
switch(config-applet)# action 4 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet5 override __pfm_tempev_major
switch(config-applet)# event temperature module 3 threshold major
switch(config-applet)# action 5 policy-default
switch(config-applet)# end
```

Configuration Examples to Override (Disable) Shutdown for Fan Tray Removal

Overriding (Disabling) a Shutdown for Removal of One or More Fan Trays

This example shows how to disable a shutdown so that you can remove one or more (or all) fan trays:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
```

Overriding (Disabling) a Shutdown for Removal of a Specified Fan Tray

This example shows how to disable a shutdown so that you can remove a specified fan tray (fan tray 3):

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 3 time 60
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config) no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config)# end
```

Overriding (Disabling) a Shutdown for Removal of Multiple Specified Fan Trays

This example shows how to disable a shutdown so that you can remove multiple specified fan trays (fan trays 2, 3, and 4):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2 time 60
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 3 time 60
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 4 time 60
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config)# end
```

Overriding (Disabling) a Shutdown for Removal of All Fan Trays Except One

This example shows how to disable a shutdown so that you can remove all fan trays except one (fan tray 2):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2 time 60
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Overriding (Disabling) a Shutdown for Removal of Fan Trays Except for a Specified Set of Fan Trays

This example shows how to disable a shutdown so that you can remove fans except for a specified set of fan trays (fan trays 2, 3, and 4):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2,3,4 time 60
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Overriding (Disabling) a Shutdown for Removal of All Fan Trays Except One from a Set of Fan Trays

This example shows how to disable a shutdown so that you can remove all fan trays except one from a set of fan trays (fan trays 2, 3, or 4):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2 time 60
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 3 time 60
switch(config-applet)# action 3 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet4 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 4 time 60
switch(config-applet)# action 4 policy-default
switch(config-applet)# end
```

Configuration Examples to Create a Supplemental Policy

Creating a Supplemental Policy for the Fan Tray Absent Event

This example shows how to create a supplemental policy using the **event fanabsent** command:

[**no**] **event fanabsent** [**fan** *fan-tray-number*] **time** *time-interval*

In addition to the default policy, this example shows how to execute the policy myappletname and action 3 if fan tray 1 is absent for 60 seconds:

```
switch# configure terminal
switch(config)# event manager applet myappletname
switch(config-applet)# event fanabsent fan 1 time 60
switch(config-applet)# action 3 cli "show env fan"
switch(config-applet)# end
```

Creating a Supplemental Policy for the Temperature Threshold Event

This example shows how to create a supplemental policy using the **event temperature** command:

[**no**] **event temperature** [**mod** *module-number*] [**sensor** *sensor-number*] **threshold** {**major** | **minor** | **any**}

In addition to the default policy, this example shows how to execute the policy myappletname and action 1 if the temperature crosses the minor threshold on sensor 3 of module 2:

```
switch# configure terminal
switch(config)# event manager applet myappletname
switch(config-applet)# event temperature module 2 sensor 3 threshold minor
switch(config-applet)# action 1 cli "show environ temperature"
switch(config-applet)# end
```

Configuration Examples for the Power Over-Budget Policy

The power over-budget policy gets triggered when the available power capacity drops below zero and the device is no longer able to keep the previously powered-up modules in the powered-up state. The default action is to print a syslog to notify the user of the occurrence of power over budget.

You can enable an additional action to power down modules until the available power recovers from the red (negative) zone.

Shutting Down Modules

If you do not specify any modules, the power over-budget shutdown starts from slot 1 and shuts down modules until the power recovers from the red (negative) zone. Empty slots and slots that contain a supervisor, standby supervisor, spine, or crossbar are skipped.

This example shows how to shut down modules starting from module 1 when the available power drops below zero:

```
switch# configure terminal
switch(config)# event manager applet <myappletname4a> override __pfm_power_over_budget
switch(config-applet)# event poweroverbudget
switch(config-applet)# action 4 overbudgetshut
switch(config-applet)# end
```

Shutting Down a Specified List of Modules

You can specify a list of modules that the power over-budget action uses to shut down modules until the power recovers from the red (negative) zone. Empty slots and slots that contain a supervisor, standby supervisor, spine, or crossbar are skipped.

This example shows how to shut down modules from a specified list of modules (1, 2, 7, 8) when the available power drops below zero:

```
switch# configure terminal
switch(config)# event manager applet <myappletname4b> override __pfm_power_over_budget
switch(config-applet)# event poweroverbudget
switch(config-applet)# action 5 overbudgetshut module 1,2,7,8
switch(config-applet)# end
```

Configuration Examples to Select Modules to Shut Down

Using the Policy Default to Select Nonoverridden Modules to Shut Down

This example shows how to use the policy default to select the nonoverridden modules to shut down when a major threshold is exceeded:

```
switch# configure terminal
switch(config)# event manager applet my5a1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet my5a2 override __pfm_tempev_major
switch(config-applet)# event temperature module 1-3 sensor 4 threshold major
switch(config-applet)# action 5 policy-default
switch(config-applet)# end
```

Using Parameter Substitution to Select Nonoverridden Modules to Shut Down

This example shows how to use parameter substitution to select the nonoverridden modules to shut down when a major threshold is exceeded:

```
switch# configure terminal
switch(config)# event manager applet my5b1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet my5b2 override __pfm_tempev_major
switch(config-applet)# event temperature module 1-3 sensor 8 threshold major
switch(config-applet)# action 6 forceshut module my_module_list reset "temperature-sensor
policy trigger"
switch(config-applet)# end
```

To create event manager parameters, use the **event manager environment** command. To display the values of event manager parameters, use the **show event manager environment all** command.

Configuration Examples for the Online Insertion Removal Event

The online insertion removal (OIR) event does not have a default policy.

This example shows how to configure the OIR event using the **event oir** command:

event oir *device-type event-type* [*device-number*]

The *device-type* can be **fan**, **module**, or **powersupply**.

The *event-type* can be **insert**, **remove**, or **anyoir** (insert or remove).

The optional *device-number* specifies a single device. If omitted, all devices are selected.

This example shows how to configure the insert event:

```
switch# configure terminal
switch(config)# event manager applet myoir
switch(config-applet)# event oir module insert
switch(config-applet)# action 1 syslog priority critical msg "OIR insert event: A Module
is inserted"
```

This example shows how to configure the remove event:

```
switch# configure terminal
switch(config)# event manager applet myoir
switch(config-applet)# event oir module remove
switch(config-applet)# action 1 syslog priority critical msg "OIR remove event: A Module
is removed"
```

Configuration Example to Generate a User Syslog

This example shows how to generate a user syslog using the **action syslog** command:

```
switch# configure terminal
switch(config)# event manager applet myoir
switch(config-applet)# event oir module remove
switch(config-applet)# action 1 syslog priority critical msg "Module is removed"
```

When this event is triggered, the system generates a syslog as follows:

```
switch(config)# 2013 May 20 00:08:27 p1b-57 %$ VDC-1 %$ %EEM_ACTION-2-CRIT: "Module is
removed"
```

Configuration Example to Monitor Syslog Messages

This example shows how to monitor syslog messages from the switch:

```
switch(config)# event manager applet a1
switch(config-applet)# event syslog occurs 6 period 4294967 pattern "authentication failed"
```

When this event is triggered, the action defined in the policy is executed.

Configuration Examples for SNMP Notification

Polling an SNMP OID to Generate an EEM Event

The SNMP object ID (OID) CISCO-SYSTEM-EXT-MIB::cseSysCPUUtilization is used for querying the CPU utilization of the switch:

```
cseSysCPUUtilization OBJECT-TYPE
SYNTAX Gauge32 (0..100 )
UNITS "%"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The average utilization of CPU on the active supervisor."
::= { ciscoSysInfoGroup 1 }
```

This example shows the use of an SNMP OID that is polled at an interval of 10 seconds and has a threshold value of 95 percent:

```
switch# configure terminal
switch(config)# event manager applet test_policy
switch(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.305.1.1.1.0 get-type exact entry-op
gt entry-val 95 exit-op lt exit-val 90 poll-interval 10
```

Sending an SNMP Notification in Response to an Event in the Event Policy

You can use this type of configuration to cause a critical event trigger to generate an SNMP notification.

This example shows how to send an SNMP notification for an event from the Event Manager applet configuration mode:

```
switch(config-applet)# action 1.1 snmp-trap intdata1 100 intdata2 300 strdata "CPU Hogging
 at switch1"
switch(config-applet)# action 1.1 snmp-trap intdata1 100 intdata2 300 strdata "Port Failure
 eth9/1"
```

This configuration triggers an SNMP notification (trap) from the switch to SNMP hosts. The SNMP payload carries the values of user-defined fields intdata1, intdata2, and strdata.

Configuration Example for Port Tracking

This example shows how to configure the state of one port to match the state of another port (port tracking).

To configure the port tracking of Ethernet interface 3/23 by Ethernet interface 1/2, follow these steps:

Procedure

Step 1 Create an object to track the status of Ethernet interface 3/23.

Example:

```
switch# configure terminal
switch(config)# track 1 interface ethernet 3/23
switch(config-track)# end
```

Step 2 Configure an EEM event to shut Ethernet interface 1/2 when the tracking object shuts down.

Example:

```
switch(config)# event manager applet track_3_23_down
switch(config-applet)# event track 1 state down
switch(config-applet)# action 1 syslog msg EEM applet track_3_23_down shutting down port
eth1/2 due to eth3/23 being down
switch(config-applet)# action 2 cli conf term
switch(config-applet)# action 3 cli interface ethernet 1/2
switch(config-applet)# action 4 cli shut
switch(config-applet)# end
```

Step 3 Configure an EEM event to bring up Ethernet interface 1/2 when Ethernet interface 3/23 comes up.

Example:

```
switch# configure terminal
switch(config)# event manager applet track_3_23_up
switch(config-applet)# event track 1 state up
switch(config-applet)# action 1 syslog msg EEM applet track_3_23_down bringing up port
eth1/2 due to eth3/23 being up
switch(config-applet)# action 2 cli conf term
switch(config-applet)# action 3 cli interface ethernet 1/2
switch(config-applet)# action 4 cli no shut
switch(config-applet)# end
```

Configuration Example to Register an EEM Policy with the EEM

This example shows how to register an EEM policy with the EEM:

Basic switch configuration:

```
event manager applet vpc_check_peer_at_startup
event track 101 state up
action 1.0 cli copy bootflash:eem/user_script_policies/load_schedules running-config

feature scheduler

!!## 2 x dummy loopbacks are required ##!!
interface loopback 101
interface loopback 102

track 1 list boolean or
```

```
object 13
object 12
object 102
track 2 list boolean and
object 13
object 12
track 12 interface Ethernet 2/24 line-protocol
track 13 interface port-channel 3000 line-protocol
track 101 interface loopback 101 line-protocol
track 102 interface loopback 102 line-protocol
```

Note In this example, port channel 3000 is the vPC peer link, and Ethernet 2/24 is the vPC keepalive link.

You need to copy the following files to the bootflash:

- A directory called: /eem/user_script_policies needs to be created on the supervisor bootflash.
- These five files need to be created and loaded into the above directory:
  - load_schedules
  - remove_vpc_if_peer_failed
  - clean_up
  - unload_schedules
  - restore_vpc

Configuration for the load_schedules file:

```
feature scheduler

configure terminal
scheduler job name vpc_check
configure terminal
event manager policy remove_vpc_if_peer_failed
end

configure terminal
scheduler job name clean_up
configure terminal
event manager policy clean_up
end

configure terminal
scheduler job name trigger
configure terminal
interface loopback 102
shutdown
no shutdown
end

configure terminal
scheduler schedule name load_vpc_check
time start +00:00:04
job name vpc_check

scheduler schedule name trigger_vpc_check
```

Embedded Event Manager System Events and Configuration Examples

```
time start +00:00:05
job name trigger

scheduler schedule name load_clean_up
time start +00:00:08
job name clean_up


scheduler schedule name trigger_clean_up
time start +00:00:10
job name trigger
```

Configuration for the remove_vpc_if_peer_failed file:

```
event manager applet remove_vpc_if_peer_failed
event track 1 state down
action 1.0 cli show run vpc > bootflash://sup-active/eem/user_script_policies/vpc_saved.cfg
action 2.0 cli show run vpc > bootflash://sup-standby/eem/user_script_policies/vpc_saved.cfg
action 3.0 cli configure terminal
action 4.0 cli no feature vpc
action 5.0 syslog msg severity alert "##### WARNING!!!! PEER SWITCH FAILED TO COME ONLINE.
 VPC CONFIG REMOVED #####"
action 6.0 cli event manager policy restore_vpc
action 7.0 cli copy bootflash:eem/user_script_policies/unload_schedules running-config
action 8.0 cli no event manager applet remove_vpc_if_peer_failed
action 9.0 cli end
```

Configuration for the clean_up file:

```
event manager applet clean_up
event track 102 state up
action 1.0 cli configure terminal
action 2.0 cli no event manager applet remove_vpc_if_peer_failed
action 3.0 cli copy bootflash:eem/user_script_policies/unload_schedules running
action 4.0 cli no event manager applet clean_up
action 5.0 end
```

Configuration for the unload_schedules file:

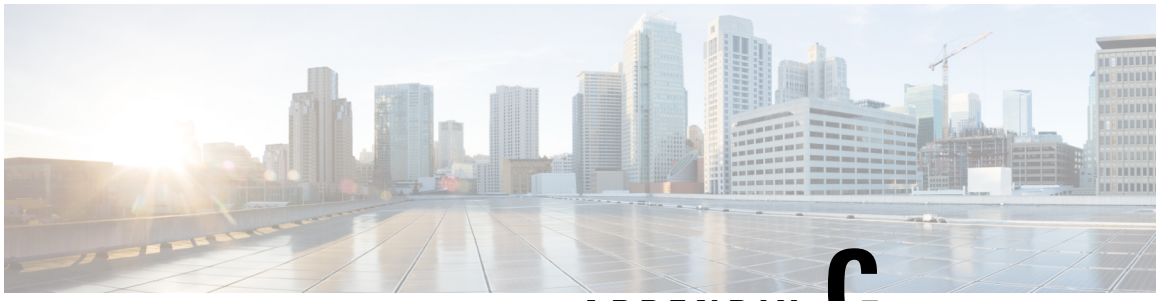
```
no scheduler schedule name load_vpc_check
no scheduler schedule name trigger_vpc_check
no scheduler schedule name load_clean_up
no scheduler schedule name trigger_clean_up
no scheduler job name vpc_check
no scheduler job name trigger
no scheduler job name clean_up
```

Configuration for the restore_vpc file:

```
event manager applet restore_vpc
event track 2 state up
action 1.0 cli copy bootflash:eem/user_script_policies/vpc_saved.cfg running-config
action 1.0 syslog priority alerts msg VPC PEER DETECTED. VPC CONFIG RESTORED
action 3.0 cli configure terminal
action 4.0 cli copy bootflash:eem/user_script_policies/unload_schedules running-config
action 5.0 cli no event manager applet restore_vpc
action 6.0 cli end
```

Note The severity keyword is deprecated and only the following patterns are allowed:
[0-9 a-zA-Z][0-9 a-zA-Z]*[-_ ,:/0-9a-zA-Z]*

Embedded Event Manager System Events and Configuration Examples



APPENDIX C

Configuration Limits for Cisco NX-OS System Management

The configuration limits are documented in the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

Configuration Limits for Cisco NX-OS System Management

The features supported by Cisco NX-OS have maximum configuration limits. Some of the features have configurations that support limits less than the maximum limits.

The configuration limits are documented in the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

Configuration Limits for Cisco NX-OS System Management