Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.2(x)

First Published: 2018-07-18

Last Modified: 2024-01-12

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.

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 9.2(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 9.2(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.2(x)

Feature	Description	Changed in Release	Where Documented
PTP	Added supported on the Cisco Nexus 9504-FM-R	9.2(3)	Guidelines and Limitations for PTP, on page 37
Configuration Replace	Updates the **configure replace** command from maintenance mode to include a user-confirmation and a warning.	9.2(2)	Guidelines and Limitations for Configuration Replace, on page 369
NetFlow	Added support for NetFlow CE on the Cisco Nexus 9300-EX platform switches	9.2(2)	Guidelines and Limitations for NetFlow, on page 286
NetFlow	Added support for Cisco Nexus 9500 platform modular switches with 9736C-EX, 97160YC-EX, 9732C-EX, and 9732C-EXM line cards.	9.2(2)	Guidelines and Limitations for NetFlow, on page 286
NetFlow	Added support for Layer 2 FEX.	9.2(1)	Guidelines and Limitations for NetFlow, on page 286
sFlow	sFlow export over VXLAN is supported.	9.2(1)	Guidelines and Limitations for sFlow, on page 300

Feature	Description	Changed in Release	Where Documented
PTP	Updated support for IEEE 802.1AS.	9.2(1)	Guidelines and Limitations for PTP, on page 37
IEEE 802.1Qaz standards for DCBXP TLV	Added Support for the Cisco Nexus 9000 Series switches.	9.2(1)	About DCBXP, on page 272
LLDP	Introduced the *show qos dcbxp interface* command.	9.2(1)	Verifying the LLDP Configuration, on page 281
System message logging	Added support to send syslog messages to remote logging servers over secure TLS transport connection.	9.2(1)	Configuring System Message Logging, on page 77
TAP aggregation filters based on MPLS tags.	Introduced in this release.	9.2(1)	Guidelines and Limitations for TAP Aggregation, on page 312

CHAPTER 2

Overview

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

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* and the *Cisco NX-OS Licensing Options Guide*.

Supported Platforms

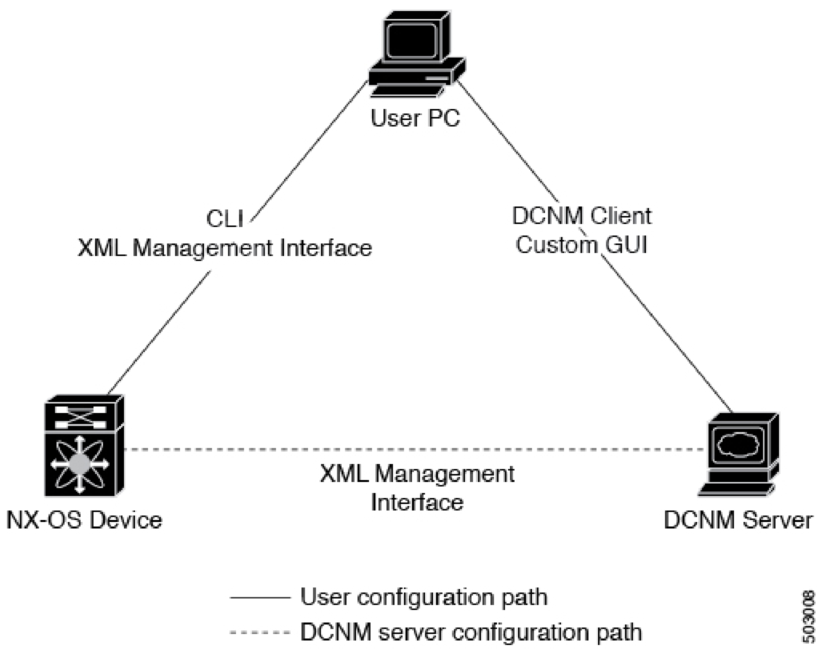
Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

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.

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.

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.

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.

MPLS Stripping

MPLS stripping provides the ability to strip MPLS labels from packets, enabling non-MPLS-capable network monitoring tools to monitor packets.

sFlow

Sampled flow (sFlow) allows you to monitor real-time traffic in data networks that contain switches and routers and to forward the sample data to a central data collector.

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 Switch Profiles

This chapter describes how to configure switch profiles on the Cisco Nexus 9000 Series switches.

About Switch Profiles

Several applications require consistent configuration across devices in the network. For example, with a virtual port channel (vPC), you must have identical configurations. Mismatched configurations can cause errors or misconfigurations that can result in service disruptions. The configuration synchronization (config-sync) feature allows you to configure one switch profile and have the configuration be automatically synchronized to the peer switch.

A switch profile provides the following benefits:

- Allows configurations to be synchronized between switches.
- Merges configurations when connectivity is established between two switches.
- Provides control of exactly which configuration gets synchronized.
- Ensures configuration consistency across peers through merge and mutual-exclusion checks.
- Provides verify and commit semantics.
- Allows for migrating existing vPC configurations to a switch profile.

Switch Profile Configuration Modes

The switch profile feature includes the following configuration modes:

- Configuration synchronization mode (config-sync)
- Switch profile mode (config-sync-sp)
- Switch profile import mode (config-sync-sp-import)

Configuration Synchronization Mode

The configuration synchronization mode (config-sync) allows you to create switch profiles.

Switch Profile Mode

The switch profile mode (config-sync-sp) allows you to add supported configuration commands to a switch profile temporary buffer that is later synchronized with a peer switch. Commands that you enter in the switch profile mode are not executed until you enter the **commit** command. Although the syntax of the commands are validated when you enter them, there is no guarantee that the commands will be successful when you enter the **commit** command.

Switch Profile Import Mode

The switch profile import mode (config-sync-sp-import) allows you to import existing switch configurations from the running configuration to a switch profile and specify which commands you want to include in that profile. This option is especially useful when you upgrade from a Cisco NX-OS release that does not support switch profiles to a release that does.

Cisco recommends that you import the necessary configurations from the running configuration using the switch profile import mode and commit the changes before making any additional changes in the switch profile or global configuration mode. Otherwise, you might jeopardize the import, requiring you to abandon the current import session and perform the process again. For more information, see Importing a Switch Profile, on page 16.

Configuration Validation

Two types of configuration validation checks can identify switch profile failures:

- Mutual exclusion checks
- Merge checks

Mutual Exclusion Checks

The mutual exclusion of configuration commands is enforced in order to avoid duplicate commands in the config-sync and global configuration modes. When you commit the configuration of a switch profile, mutual exclusion (mutex) checks are performed on the local switch as well as the peer switch (if configured). If no failures are reported on both switches, the commit is accepted and pushed into the running configuration.

A command that is included in a switch profile cannot be configured outside of the switch profile.

If a mutex check identifies errors, they are reported as mutex failures, and they must be manually corrected. For details, see Manually Correcting Mutex and Merge Failures, on page 20.

The following exceptions apply to the mutual exclusion policy:

- Interface configuration—An interface configuration can be partially present in a switch profile and partially present in the running configuration as long as there are no conflicts.
- Shutdown/no shutdown
- System QoS

Merge Checks

Merge checks are done on the peer switch that is receiving a configuration. The merge checks ensure that the received configuration does not conflict with the switch profile configuration that already exists on the receiving switch. The merge check occurs during the verify or commit process. Errors are reported as merge failures and must be manually corrected. For details, see Manually Correcting Mutex and Merge Failures, on page 20.

When one or both switches are reloaded and the configurations are synchronized for the first time, the merge check verifies that the switch profile configurations are identical on both switches. Differences in the switch profiles are reported as merge errors and must be manually corrected.

Software Upgrades and Downgrades with Switch Profiles

You must delete the switch profile when downgrading from a Cisco NX-OS release that supports switch profiles to a release that does not.

When you upgrade from an earlier release to a Cisco NX-OS release that supports switch profiles, you have the option to move some of the running-configuration commands to a switch profile. For details, see Switch Profile Import Mode, on page 10.

An upgrade can occur if there are buffered (uncommitted) configurations; however, the uncommitted configurations will be lost.

Guidelines and Limitations for Switch Profiles

Switch profiles have the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(3), the **mtu** command is supported in the interface configuration mode through the switch-profiles configuration mode.
- Switch profiles are supported only on Cisco Nexus 9300 Series switches. Cisco Nexus 9500 Series switches do not support switch profiles.
- You can only enable configuration synchronization using the mgmt0 interface.
- When using config-sync in a virtual peer-link environment, note the following limitations:
  - To initiate a config-sync session with a virtual peer link, be sure to configure a loopback IP address instead of a management IP address between the peer switches.
  - You cannot perform a configuration synchronization between a multichassis EtherChannel trunk (MCT) configuration and a virtual peer-link configuration. This config-sync operation is not supported.

- You must configure synchronized peers with the same switch profile name.
- Commands that are qualified for a switch profile configuration are allowed to be configured in the configuration switch profile mode (config-sync-sp).
- Supported switch profile commands relate to vPC commands.
- Only one switch profile session can be in progress at a time. Attempts to start another session will fail.
- Command changes made from the global configuration mode are blocked when a switch profile session is in progress.
- When you enter the **commit** command and a peer switch is reachable, the configuration is applied to both peer switches or neither switch. If a commit failure occurs, the commands remain in the switch profile buffer. You can then make necessary corrections and try the commit again.
- The configuration synchronization (**config-sync**) mode is an L2 mode parallel to the config-terminal mode (**config t**). Config-sync uses the switch-profile to update **config t** mode in the same switch as well as the peer switch. To prevent sync issues in **switch-profile** mode, Cisco recommends that you perform a commit action after each CLI command before overriding, or replacing the current CLI command.

  For example, if you want to overwrite **CLI_command_A** and change it to **CLI_command_B**, commit **CLI_command_A** first, then configure **CLI_command_B** and perform another commit action.

```
switch# conf sync
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-sync)# switch-profile test
Resyncing db before starting Switch-profile.Re-synchronization of switch-profile db
takes a few minutes...
Re-synchronize switch-profile db completed successfully.
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)#
switch(config-sync-sp)# int e 1/3
switch(config-sync-sp-if)# switchport trunk allowed vlan 100-150
switch(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch(config-sync)#
switch(config-sync)# switch-profile test
Resyncing db before starting Switch-profile.Re-synchronization of switch-profile db
takes a few minutes...
Re-synchronize switch-profile db completed successfully.
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)#
switch(config-sync-sp)# int e 1/3
switch(config-sync-sp-if)# switchport trunk allowed vlan 45-90
switch(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch(config-sync)# end
switch#
```

- Layer 3 commands are not supported.

Configuring Switch Profiles

You can create and configure a switch profile on the local switch and then add a second switch that will be included in the synchronization.

You must create the switch profile with the same name on each switch, and the switches must configure each other as a peer. When connectivity is established between switches with the same active switch profile, the switch profiles are synchronized.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Required: **cfs ipv4 distribute**

Example:

```
switch(config)# cfs ipv4 distribute
```

Enables Cisco Fabric Services (CFS) distribution between the peer switches.

Step 3 Required: **config sync**

Example:

```
switch(config)# config sync
switch(config-sync)#
```

Enters the configuration synchronization mode.

Step 4 Required: **switch-profile** *name*

Example:

```
switch(config-sync)# switch-profile abc
switch(config-sync-sp)#
```

Configures the switch profile, names the switch profile, and enters the switch profile configuration mode.

Step 5 Required: [**no**] **sync-peers destination** *ip-address*

Example:

```
switch(config-sync-sp)# sync-peers destination 10.1.1.1
```

Adds a switch to the switch profile. The destination IP address is the IP address of the switch that you want to synchronize.

The **no** form of this command removes the specified switch from the switch profile.

Note You need to wait for peer switches to show the switch-profile status of "In sync" before any commit is done.

Step 6 Required: For Cisco Nexus 3164Q switches only, follow these steps:

a) **interface** *type slot/port*

Example:

```
switch(config-sync-sp)# interface ethernet 1/1
switch(config-sync-sp-if)#
```

Enters the switch profile interface configuration mode.

b) **switchport**

Example:

```
switch(config-sync-sp-if)# switchport
```

Changes a Layer 3 interface into a Layer 2 interface.

c) **exit**

Example:

```
switch(config-sync-sp-if)# exit
switch(config-sync-sp)#
```

Exits the switch profile interface configuration mode.

d) **commit**

Example:

```
switch(config-sync-sp)# commit
```

Commits the current configuration.

Note Verify that the switch-profile status shows as "In sync" before any commit is done.

Step 7 (Optional) **end**

Example:

```
switch(config-sync-sp)# end
switch#
```

Exits the switch profile configuration mode and returns to EXEC mode.

Step 8 (Optional) **show switch-profile** *name* **status**

Example:

```
switch# show switch-profile abc status
```

Displays the switch profile on the local switch and the peer switch information.

Step 9 (Optional) **show switch-profile** *name* **peer** *ip-address*

Example:

```
switch# show switch-profile abc peer 10.1.1.1
```

Displays the switch profile peer configuration.

Step 10 (Optional) **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

Adding or Modifying Switch Profile Commands

After you configure a switch profile on the local and the peer switch, you must add and commit the supported commands to the switch profile.

Commands that are added or modified are buffered until you enter the **commit** command. Commands are executed in the same order in which they are buffered. If there is an order dependency for certain commands (for example, a QoS policy must be defined before being applied), you must maintain that order; otherwise, the commit might fail. You can use utility commands, such as the **show switch-profile** *name* **buffer** command, the **buffer-delete** command, and the **buffer-move** command, to change the buffer and correct the order of already entered commands.

Procedure

	Command or Action	Purpose

Step 1	Required: **config sync**
	Example:
	`switch# config sync`
	`switch(config-sync)#`
		Enters the configuration synchronization mode.

Step 2	Required: **switch-profile** *name*
	Example:
	`switch(config-sync)# switch-profile abc`
	`switch(config-sync-sp)#`
		Configures the switch profile, names the switch profile, and enters the switch profile configuration mode.

Step 3	Required: *command*
	Example:
	`switch(config-sync-sp)# interface Port-channel100`
	`switch(config-sync-sp-if)# speed 1000`
	`switch(config-sync-sp-if)# interface Ethernet1/1`
	`switch(config-sync-sp-if)# speed 1000`
	`switch(config-sync-sp-if)# channel-group 100`
	`switch(config-sync-sp-if)# exit`
	`switch(config-sync-sp)#`
		Adds a command to the switch profile.

Step 4	(Optional) **show switch-profile** *name* **buffer** **Example:** `switch(config-sync-sp)# show switch-profile abc buffer`	Displays the configuration commands in the switch profile buffer.
Step 5	Required: **verify** **Example:** `switch(config-sync-sp)# verify`	Verifies the commands in the switch profile buffer.

	Command or Action	Purpose
Step 6	Required: **commit** **Example:** `switch(config-sync-sp)# commit`	Saves the commands in the switch profile and synchronizes the configuration with the peer switch. This command also does the following: • Triggers the mutex check and the merge check to verify the synchronization. • Creates a checkpoint with a rollback infrastructure. • Executes a rollback on all switches if an application failure occurs on any of the switches in the switch profile. • Deletes the checkpoint.

Step 7	(Optional) **end**
	Example:
	`switch(config-sync-sp)# end`
	`switch#`
		Exits the switch profile configuration mode and returns to EXEC mode.

Step 8	(Optional) **show switch-profile** *name* **status** **Example:** `switch# show switch-profile abc status`	Displays the status of the switch profile on the local switch and the status on the peer switch.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Importing a Switch Profile

You can import a switch profile based on the set of commands that you want to import.

Before you begin

Make sure that the switch profile buffer is empty before you import commands to a switch profile.

Procedure

	Command or Action	Purpose

Step 1	(Optional) Configure the interface that will be imported in Step 4.
	Example:
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)# switchport`
	`switch(config-if)# switchport mode trunk`
		Enters configuration synchronization mode.

	Command or Action	Purpose

	`switch(config-if)# switchport trunk allowed vlan 12`
	`switch(config-if)# speed 10000`
	`switch(config-if)# spanning-tree port type edge trunk`
	`switch(config)# end`
	`switch#`	

Step 2	**config sync**
	Example:
	`switch# config sync`
	`switch(config-sync)#`
		Enters configuration synchronization mode.

Step 3	Required: **switch-profile** *name*
	Example:
	`switch(config-sync)# switch-profile abc`
	`switch(config-sync-sp)#`
		Configures the switch profile, names the switch profile, and enters the switch profile configuration mode.

Step 4	Required: **import** [**interface** *interface port*/*slot* | **running-config**]
	Example:
	`switch(config-sync-sp)# import interface ethernet 1/2`
	`switch(config-sync-sp-import)#`
		Identifies the commands that you want to import and enters the switch profile import mode. The following options are available:
		• Entering the **import** command without any options adds the selected commands to the switch profile.
		• The **import interface** option adds the supported commands for a specified interface.
		• The **running-config** option adds supported system-level commands.
		Note If new commands are added during the import, the switch profile remains unsaved, and the switch remains in the switch profile import mode.

Step 5	Required: **commit** **Example:** `switch(config-sync-sp-import)# commit`	Imports the commands and saves the commands to the switch profile.
Step 6	(Optional) **abort** **Example:** `switch(config-sync-sp-import)# abort`	Aborts the import process.

Step 7	(Optional) **end**
	Example:
	`switch(config-sync-sp-import)# end`
	`switch#`
		Exits the switch profile import mode and returns to EXEC mode.

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

Importing Configurations in a vPC Topology

You can import configurations in a two-switch vPC topology.

> **Note** For specific information on the following steps, see the appropriate sections in this chapter.

1. Configure the switch profile with the same name on both switches.
2. Import the configurations to both switches independently.

   > **Note** Make sure that the configuration moved to the switch profile on both switches is identical; otherwise, a merge-check failure might occur.

3. Configure the switches by entering the **sync-peers destination** command.
4. Verify that the switch profiles are the same by entering the appropriate **show** commands.

Isolating a Peer Switch

You can isolate a peer switch in order to make changes to a switch profile. This process can be used when you want to block configuration synchronization, debug configurations, or recover from a situation when the config-sync feature becomes out of sync.

Isolating a peer switch requires that you break the peer connection from the switch profile and then add the peer switch back to the switch profile.

> **Note** For specific information on the following steps, see the appropriate sections in this chapter.

1. Remove the peer switch from the switch profile on both switches.
2. Add the **no sync-peers destination** command to the switch profile and commit the changes on both switches.

3. Add any necessary troubleshooting configurations.
4. Verify that the show running switch-profile is identical on both switches.
5. Add the **sync-peers destination** *ip-address* command to both switches and commit the changes.
6. Verify that the peers are in sync.

Deleting a Switch Profile

You can delete a switch profile.

Procedure

	Command or Action	Purpose

Step 1	**config sync**
	Example:
	`switch# config sync`
	`switch(config-sync)#`
		Enters configuration synchronization mode.

Step 2	Required: **no switch-profile** *name* {**all-config** | **local-config**}
	Example:
	`switch(config-sync)# no switch-profile abc local-config`
	`switch(config-sync-sp)#`
		Deletes the switch profile as follows:
		• **all-config**—Deletes the switch profile on the local and the peer switch. If the peer switch is not reachable, only the local switch profile is deleted.
		• **local-config**—Deletes the switch profile and local configuration.
		Note It is recommended that you execute **resync-database** prior to deleting a switch-profile:
		`switch(config-sync)# resync-database`

Step 3	(Optional) **end**
	Example:
	`switch(config-sync-sp)# end`
	`switch#`
		Exits the switch profile configuration mode and returns to EXEC mode.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration. When you enter this command, the config-sync feature triggers the same operation on the peer switch.

Manually Correcting Mutex and Merge Failures

You can manually correct mutex and merge failures when they occur.

Note If the conflict is on the peer switch, follow the steps in Isolating a Peer Switch, on page 18 to correct the problem on that switch.

1. Import the offending command into the switch profile using the switch profile import mode.
2. Change the behavior as desired.

Verifying the Switch Profile Configuration

To display information about a switch profile, perform one of the following tasks:

Command	Purpose
show switch-profile *name*	Displays the commands in a switch profile.
show switch-profile *name* **buffer**	Displays the uncommitted commands in a switch profile, the commands that were moved, and the commands that were deleted.
show switch-profile *name* **peer** *ip-address*	Displays the synchronization status for a peer switch.
show switch-profile *name* **session-history**	Displays the status of the last 20 switch profile sessions.
show switch-profile *name* **status**	Displays the configuration synchronization status of a peer switch.
show running-config switch-profile	Displays the running configuration for the switch profile on the local switch.
show startup-config switch-profile	Displays the startup configuration for the switch profile on the local switch.

Configuration Examples for Switch Profiles

Creating a Switch Profile on a Local and a Peer Switch

The following example shows how to create a successful switch profile configuration on a local and a peer switch, including configuring QoS policies, a vPC peer link, and a vPC in a switch profile.

1. Enable CFS distribution on the local and the peer switch and configure the destination IP address of the switch that you want to synchronize with, such as the management interface on the switch.

```
—Local switch-1#---
switch-1# configure terminal
switch-1(config)# cfs ipv4 distribute
switch-1(config)# interface mgmt 0
switch-1(config-if)# ip address 30.0.0.81/8

—Peer switch-2#--
switch-2# configure terminal
switch-2(config)# cfs ipv4 distribute
switch-2(config)# interface mgmt 0
switch-2(config-if)# ip address 30.0.0.82/8
```

2. Create a new switch profile on the local and the peer switch.

```
—Local switch-1#---
switch-1# config sync
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# sync-peers destination 30.0.0.82
switch-1(config-sync-sp)# end

—Peer switch-2#--
switch-1# config sync
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# sync-peers destination 30.0.0.81
switch-1(config-sync-sp)# end
```

3. Verify that the switch profiles are the same on the local and the peer switch.

```
switch-1(config-sync-sp)# show switch-profile status

switch-profile : A
----------------------------------------------------------

Start-time: 843992 usecs after Wed Aug 19 17:00:01 2015
End-time: 770051 usecs after Wed Aug 19 17:00:03 2015

Profile-Revision: 1
Session-type: Initial-Exchange
Session-subtype: Init-Exchange-All
Peer-triggered: Yes
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 30.0.0.82
Sync-status: In sync
Status: Commit Success
Error(s):
```

4. Add the configuration commands to the switch profile on the local switch. The commands will be applied to the peer switch when the commands are committed.

```
switch-1# config sync
switch-1(config-sync)# switch-profile A
```

```
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# interface port-channel 10
switch-1(config-sync-sp-if)# switchport
switch-1(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# interface port-channel 10
switch-1(config-sync-sp-if)# switchport mode trunk
switch-1(config-sync-sp-if)# switchport trunk allowed vlan 10
switch-1(config-sync-sp-if)# spanning-tree port type network
switch-1(config-sync-sp-if)# vpc peer-link
switch-1(config-sync-sp-if)# switch-profile switching-mode switchname
switch-1(config-sync-sp-if)# show switch-profile buffer

switch-profile : A
----------------------------------------------------------
Seq-no Command
----------------------------------------------------------
1 interface port-channel10
1.1 switchport mode trunk
1.2 switchport trunk allowed vlan 10
1.3 spanning-tree port type network
1.4 vpc peer-link

switch-1(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# interface ethernet 2/1
switch-1(config-sync-sp-if)# switchport mode trunk
switch-1(config-sync-sp-if)# switchport trunk allowed vlan 10
switch-1(config-sync-sp-if)# spanning-tree port type network
switch-1(config-sync-sp-if)# channel-group 10 mode active
```

5. View the buffered commands.

```
switch-1(config-sync-sp-if)# show switch-profile buffer

switch-profile : A
----------------------------------------------------------
Seq-no Command
----------------------------------------------------------
1 interface Ethernet2/1
1.1 switchport mode trunk
1.2 switchport trunk allowed vlan 10
1.3 spanning-tree port type network
1.4 channel-group 10 mode active
```

6. Verify the commands in the switch profile.

```
switch-1(config-sync-sp-if)# verify
Verification Successful
```

7. Apply the commands to the switch profile and synchronize the configurations between the local and the peer switch.

```
—Local switch-2#--
switch-1(config-sync-sp)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch-1(config-sync)# end

switch-1# show running-config switch-profile

switch-profile A
sync-peers destination 30.0.0.82

interface port-channel10
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
vpc peer-link

interface Ethernet2/1
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
channel-group 10 mode active


—Peer switch-2#--
switch-2# show running-config switch-profile

switch-profile A
sync-peers destination 30.0.0.81

interface port-channel10
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
vpc peer-link

interface Ethernet2/1
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
channel-group 10 mode active
```

Verifying the Synchronization Status

The following example shows how to verify the synchronization status between the local and the peer switch:

```
switch-1# show switch-profile status

switch-profile : A
-------------switch-1---------------------------------------------

Start-time: 912776 usecs after Wed Aug 19 17:03:43 2015
End-time: 868379 usecs after Wed Aug 19 17:03:48 2015

Profile-Revision: 4
```

```
Session-type: Commit
Session-subtype: -
Peer-triggered: No
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 30.0.0.82
Sync-status: In sync
Status: Commit Success
Error(s):
```

Showing the Running Configuration

The following example shows the running configuration of the switch profile on the local switch:

```
—— PEER SWITCH-1 ——
switch-1# show running-config switch-profile

switch-profile A
 sync-peers destination 30.0.0.82

 interface port-channel10
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 vpc peer-link

 interface Ethernet2/1
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 channel-group 10 mode active
switch-1#

—— PEER SWITCH-2 ——
switch-2# show running-config switch-profile

switch-profile A
 sync-peers destination 30.0.0.81

 interface port-channel10
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 vpc peer-link

 interface Ethernet2/1
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 channel-group 10 mode active
switch-2#
```

Displaying the Switch Profile Synchronization Between the Local and the Peer Switch

The following example shows how to display the initial successful synchronization between the two peers:

```
switch1# show switch-profile sp status

Start-time: 491815 usecs after Mon Jul 20 11:54:51 2015
End-time: 449475 usecs after Mon Jul 20 11:54:58 2015

Profile-Revision: 1
Session-type: Initial-Exchange
Peer-triggered: No
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.52
Sync-status: In Sync.
Status: Commit Success
Error(s):


switch2# show switch-profile sp status

Start-time: 503194 usecs after Mon Jul 20 11:54:51 2015
End-time: 532989 usecs after Mon Jul 20 11:54:58 2015

Profile-Revision: 1
Session-type: Initial-Exchange
Peer-triggered: Yes
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.51
Sync-status: In Sync.
Status: Commit Success
Error(s):
```

Displaying Verify and Commit on the Local and the Peer Switch

The following example shows how to perform a successful verify and commit of the local and the peer switch:

```
switch1# config sync
switch1(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1
switch1(config-sync-sp)# interface Ethernet1/1
switch1(config-sync-sp-if)# description foo
switch1(config-sync-sp-if)# exit
```

```
switch1(config-sync-sp)# verify
Verification Successful
switch1(config-sync-sp)# commit
Commit Successful
switch1(config-sync)# show running-config switch-profile
switch-profile sp
  sync-peers destination 10.193.194.52
  interface Ethernet1/1
    description foo
switch1(config-sync)# show switch-profile sp status

Start-time: 171513 usecs after Wed Jul 20 17:51:28 2015
End-time: 676451 usecs after Wed Jul 20 17:51:43 2015

Profile-Revision: 3
Session-type: Commit
Peer-triggered: No
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.52
Sync-status: In Sync.
Status: Commit Success
Error(s):

switch1(config-sync)#


switch2# show running-config switch-profile
switch-profile sp
  sync-peers destination 10.193.194.51
  interface Ethernet1/1
    description foo
switch2# show switch-profile sp status

Start-time: 265716 usecs after Mon Jul 20 16:51:28 2015
End-time: 734702 usecs after Mon Jul 20 16:51:43 2015

Profile-Revision: 3
Session-type: Commit
Peer-triggered: Yes
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.51
Sync-status: In Sync.
Status: Commit Success
Error(s):
```

Displaying the Successful and Unsuccessful Synchronization Between the Local and the Peer Switch

The following example shows how to configure the synchronization status of the switch profile on the peer switch. The first example shows a successful synchronization, and the second example shows a peer-not-reachable status.

```
switch1# show switch-profile sp peer

switch1# show switch-profile sp peer 10.193.194.52
Peer-sync-status           : In Sync.
Peer-status                : Commit Success
Peer-error(s)              :
switch1#


switch1# show switch-profile sp peer 10.193.194.52
Peer-sync-status           : Not yet merged. pending-merge:1 received_merge:0
Peer-status                : Peer not reachable
Peer-error(s)              :
```

Displaying the Switch Profile Buffer

The following example shows how to configure the switch profile buffer, the buffer-move configuration, and the buffer-delete configuration:

```
switch1# config sync
switch1(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1
switch1(config-sync-sp)# vlan 101
switch1(config-sync-sp-vlan)# ip igmp snooping querier 10.101.1.1
switch1(config-sync-sp-vlan)# exit
switch1(config-sync-sp)# mac address-table static 0000.0000.0001 vlan 101 drop
switch1(config-sync-sp)# interface Ethernet1/2
switch1(config-sync-sp-if)# switchport mode trunk
switch1(config-sync-sp-if)# switchport trunk allowed vlan 101
switch1(config-sync-sp-if)# exit
switch1(config-sync-sp)# show switch-profile sp buffer
----------------------------------------------------------
Seq-no  Command
----------------------------------------------------------
1       vlan 101
1.1       ip igmp snooping querier 10.101.1.1
2       mac address-table static 0000.0000.0001 vlan 101 drop
3       interface Ethernet1/2
3.1       switchport mode trunk
3.2       switchport trunk allowed vlan 101

switch1(config-sync-sp)# buffer-move 3 1
switch1(config-sync-sp)# show switch-profile sp buffer
----------------------------------------------------------
Seq-no  Command
----------------------------------------------------------
1       interface Ethernet1/2
1.1       switchport mode trunk
1.2       switchport trunk allowed vlan 101
2       vlan 101
2.1       ip igmp snooping querier 10.101.1.1
3       mac address-table static 0000.0000.0001 vlan 101 drop
```

```
switch1(config-sync-sp)# buffer-delete 1
switch1(config-sync-sp)# show switch-profile sp buffer
----------------------------------------------------------
Seq-no  Command
----------------------------------------------------------
2       vlan 101
2.1       ip igmp snooping querier 10.101.1.1
3       mac address-table static 0000.0000.0001 vlan 101 drop

switch1(config-sync-sp)# buffer-delete all
switch1(config-sync-sp)# show switch-profile sp buffer
```

Importing Configurations

The following example shows how to import an interface configuration:

```
switch# show running-config interface Ethernet1/3

!Command: show running-config interface Ethernet1/3
!Time: Wed Jul 20 18:12:44 2015

version 7.0(3)I2(1)

interface Ethernet1/3
  switchport mode trunk
  switchport trunk allowed vlan 1-100

switch# config sync
switch(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1

switch(config-sync-sp)# import interface Ethernet1/3
switch(config-sync-sp-import)# show switch-profile sp buffer
----------------------------------------------------------
Seq-no  Command
----------------------------------------------------------
1       interface Ethernet1/3
1.1       switchport mode trunk
1.2       switchport trunk allowed vlan 1-100

switch(config-sync-sp-import)# verify
Verification Successful
switch(config-sync-sp-import)# commit
Commit Successful
```

The following example shows how to import the supported commands in a running configuration:

```
switch(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)# import running-config
switch(config-sync-sp-import)# show switch-profile sp buffer
----------------------------------------------------------
Seq-no  Command
----------------------------------------------------------
1       logging event link-status default
2       vlan 1
3       interface port-channel 3
3.1       switchport mode trunk
3.2       vpc peer-link
3.3       spanning-tree port type network
```

```
4       interface port-channel 30
4.1       switchport mode trunk
4.2       vpc 30
4.3       switchport trunk allowed vlan 2-10
5       interface port-channel 31
5.1       switchport mode trunk
5.2       vpc 31
5.3       switchport trunk allowed vlan 11-20
6       interface port-channel 101
6.1       switchport mode fex-fabric
6.2       fex associate 101
7       interface port-channel 102
7.1       switchport mode fex-fabric
7.2       vpc 102
7.3       fex associate 102
8       interface port-channel 103
8.1       switchport mode fex-fabric
8.2       vpc 103
8.3       fex associate 103
9      interface Ethernet1/1
10      interface Ethernet1/2
11      interface Ethernet1/3
12      interface Ethernet1/4
12.1      switchport mode trunk
12.2      channel-group 3
13      interface Ethernet1/5
13.1      switchport mode trunk
13.2      channel-group 3
14      interface Ethernet1/6
14.1      switchport mode trunk
14.2      channel-group 3
15      interface Ethernet1/7
15.1      switchport mode trunk
15.2      channel-group 3
16      interface Ethernet1/8
17      interface Ethernet1/9
17.1      switchport mode trunk
17.2      switchport trunk allowed vlan 11-20
17.3      channel-group 31 mode active
18      interface Ethernet1/10
18.1      switchport mode trunk
18.2      switchport trunk allowed vlan 11-20
18.3      channel-group 31 mode active
19      interface Ethernet1/11
20      interface Ethernet1/12
...
45      interface Ethernet2/4
45.1      fex associate 101
45.2      switchport mode fex-fabric
45.3      channel-group 101
46      interface Ethernet2/5
46.1      fex associate 101
46.2      switchport mode fex-fabric
46.3      channel-group 101
47      interface Ethernet2/6
47.1      fex associate 101
47.2      switchport mode fex-fabric
47.3      channel-group 101
48      interface Ethernet2/7
48.1      fex associate 101
48.2      switchport mode fex-fabric
48.3      channel-group 101
49      interface Ethernet2/8
49.1      fex associate 101
```

```
...
89      interface Ethernet100/1/32
90      interface Ethernet100/1/33
91      interface Ethernet100/1/34
92      interface Ethernet100/1/35
93      interface Ethernet100/1/36
...
105     interface Ethernet100/1/48
```

Migrating to Cisco NX-OS Release 7.0(3)I2(1) or Higher in a Fabric Extender Straight-Through Topology

This example shows the tasks used to migrate to Cisco NX-OS Release 7.0(3)I2(1) or higher in a Fabric Extender active/active or straight-through topology. For details on the tasks, see the appropriate sections in this chapter.

1. Make sure configurations are the same on both switches.
2. Configure the switch profile with the same name on both switches.
3. Enter the **import interface port-channel** *x-y*, **port-channel** *z* command for all vPC port channels on both switches.
4. Enter the **show switch-profile** *name* **buffer** command to ensure all configurations are correctly imported on both switches.
5. Remove unwanted configuration settings by editing the buffer.
6. Enter the **commit** command on both switches.
7. Enter the **sync-peers destination** *ip-address* command to configure the peer switch on both switches.
8. Enter the **show switch-profile** *name* **status** command to ensure both switches are synchronized.

Replacing a Cisco Nexus 9000 Series Switch

When a Cisco Nexus 9000 Series switch has been replaced, perform the following configuration steps on the replacement switch to synchronize it with the existing Cisco Nexus 9000 Series switch. This procedure can be done in a hybrid Fabric Extender active/active topology and Fabric Extender straight-through topology.

1. Do not connect any peer link, vPC, active/active, or straight-through topology fabric ports to the replacement switch.
2. Boot the replacement switch. The switch comes up with no configuration.
3. Configure the replacement switch:
    - If the running configuration was saved offline, follow Steps 4 through 8 to apply the configuration.
    - If the running configuration was not saved offline, you can obtain it from the peer switch if the configuration synchronization feature is enabled. (See Steps 1 and 2 in Creating a Switch Profile on a Local and a Peer Switch, on page 20; then begin with Step 9 below).
    - If neither condition is met, manually add the configuration and then begin with Step 9 below.

4. Edit the configuration file to remove the **sync-peer** command if you are using the configuration synchronization feature.
5. Configure the mgmt port IP address and download the configuration file.
6. Copy the saved configuration file to the running configuration.
7. Verify that the configuration is correct by entering the **show running-config** command.
8. If the switch profile configuration changes were made on the peer switch while the replacement switch was out of service, apply those configurations in the switch profile and then enter the **commit** command.
9. Shut down all Fabric Extender straight-through topology ports that are included in a vPC topology.
10. Connect the Fabric Extender straight-through topology fabric ports.
11. Wait for the Fabric Extender straight-through topology switches to come online.
12. Make sure that the vPC role priority of the existing switch is better than the replacement switch.
13. Connect the peer-link ports to the peer switch.
14. Connect the switch vPC ports.
15. Enter the **no shutdown** command on all Fabric Extender straight-through vPC ports.
16. Verify that all vPC switches and the Fabric Extenders on the replacement switch come online and that there is no disruption in traffic.
17. If you are using the configuration synchronization feature, add the sync-peer configuration to the switch profile if it was not enabled in Step 3.
18. If you are using the configuration synchronization feature, enter the **show switch-profile** *name* **status** command to ensure both switches are synchronized.

Synchronizing Configurations

Synchronizing Configurations After a Cisco Nexus 9000 Series Switch Reboots

If a Cisco Nexus 9000 Series switch reboots while a new configuration is committed on a peer switch using a switch profile, follow these steps to synchronize the peer switches after the reload:

1. Remove the peer switch from the switch profile on both switches.
2. Add the **no sync-peers destination** command to the switch profile and commit the changes on both switches.
3. Add any missing or changed commands.
4. Verify that the show running switch-profile is identical on both switches.
5. Add the **sync-peers destination** *ip-address* command to both switches and commit the changes.
6. Verify that the peers are in sync.

Synchronizing Configurations When the mgmt0 Interface Connectivity Is Lost

When the mgmt0 interface connectivity is lost and configuration changes are required, apply the configuration changes on both switches using the switch profile. When connectivity to the mgmt0 interface is restored, both switches are synchronized.

If a configuration change is made on only one switch in this scenario, a merge will succeed when the mgmt0 interface comes up and the configuration gets applied on the other switch.

Reverting an Inadvertent Port Mode Change of Layer 2 to Layer 3 in Global Configuration Mode

The configurations related to a port imported in config-sync mode should never be configured in the global configuration mode. Normally any attempt to do so will be denied by the config-sync feature, and a mutex warning will appear. However, due to limitations in mutex checks, if a port configured as Layer 2 in the config-sync mode is changed to Layer 3 (no switchport) in the global configuration mode, the config-sync feature is unable to detect and prevent it. As a result, the config-sync mode might become out of sync with the global configuration mode. In this case, follow these steps to revert the change:

1. Remove the peer switch from the switch profile on both switches.
2. Add the **no sync-peers destination** command to the switch profile and commit the changes on both switches.
3. Import the current interface configuration.
4. Make any necessary changes and commit them.
5. Verify that the show running switch-profile is identical on both switches.
6. Add the **sync-peers destination** *ip-address* command to both switches and commit the changes.
7. Verify that the peers are in sync.

CHAPTER 4

Configuring PTP

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

This chapter includes the following sections:

About PTP

PTP is a time synchronization protocol defined in IEEE 1588 for nodes distributed across a network. With PTP, it is possible to synchronize distributed clocks with an accuracy of less than 1 microsecond via Ethernet networks. In addition, PTP's hardware timestamping feature provides timestamp information in the ERSPAN Type III header that can be used to calculate packet latency among edge, aggregate, and core switches.

A PTP system can consist of a combination of PTP and non-PTP devices. PTP devices include ordinary clocks, boundary clocks, and transparent clocks. Non-PTP devices include ordinary network switches, routers, and other infrastructure devices.

PTP is a distributed protocol that specifies how real-time PTP clocks in the system synchronize with each other. These clocks are organized into a master-slave synchronization hierarchy with the grandmaster clock, which is the clock at the top of the hierarchy, determining the reference time for the entire system. Synchronization is achieved by exchanging PTP timing messages, with the members using the timing information to adjust their clocks to the time of their master in the hierarchy. PTP operates within a logical scope called a PTP domain.

PTP supports the following functionality:

- Multicast and unicast PTP transport—In the multicast transport mode, PTP uses multicast destination IP address 224.0.1.129 as per IEEE 1588 standards for communication between devices. For the source

IP address, it uses the user configurable global IP address under the PTP domain. In the unicast transport mode, PTP uses configurable unicast source and destination IP addresses that can be configured under an interface. In both, the unicast and the multicast modes, PTP uses UDP ports, 319 for event messages and 320 for general messages communication between devices.

- PTP multicast configuration is supported only under physical interface for L2 or L3. Unicast PTP configuration supported only under L3 physical interface. PTP is not supported for virtual interfaces such as Port-channel, SVI, and tunnel.
- PTP encapsulation over UDP over IP—PTP uses UDP as the transport protocol over IP. In both, the unicast and multicast modes, PTP uses UDP ports 319 for event messages and 320 for general messages communication between devices. L2 encapsulation mode is not supported.
- PTP profiles—PTP supports default (1588) , AES67, and SMPTE 2059-2 profiles. They all have different ranges of sync and delay request intervals. For information on the default profile, refer to IEEE 1588. For more information on AES67 and SMPTE 2059-2, refer to the respective specifications.
- Path delay measurement—We support delay request and response mechanism to measure the delay between the master and slave devices. Peer delay request and response mechanism is not supported.
- Message intervals—You can configure the interval at which the announce, sync, and delay request messages needs to be sent between devices.
- Best master clock (BMC) selection—BMC algorithm is used to select master, slave, and passive states of the PTP enabled interfaces based on the Announce message received as per 1588 specification.

PTP Device Types

The PTP device type is configurable and can be used to set the clock type.

Clocks

The following clocks are common PTP devices:

Ordinary clock

Communicates with the network based on a single physical port, similar to an end host. An ordinary clock can function as a grandmaster clock.

Boundary clock

Typically has several physical ports, with each port behaving like a port of an ordinary clock. However, each port shares the local clock, and the clock data sets are common to all ports. Each port decides its individual state, either master (synchronizing other ports connected to it) or slave (synchronizing to a downstream port), based on the best clock available to it through all of the other ports on the boundary clock. Messages related to synchronization and establishing the master-slave hierarchy terminate in the protocol engine of a boundary clock and are not forwarded.

Transparent clock

Forwards all PTP messages like an ordinary switch or router but measures the residence time of a packet in the switch (the time that the packet takes to traverse the transparent clock) and in some cases the link delay of the ingress port for the packet. The ports have no state because the transparent clock does not need to synchronize to the grandmaster clock.

There are two kinds of transparent clocks:

End-to-end transparent clock

Measures the residence time of a PTP message and accumulates the times in the correction field of the PTP message or an associated follow-up message.

Peer-to-peer transparent clock

Measures the residence time of a PTP message and computes the link delay between each port and a similarly equipped port on another node that shares the link. For a packet, this incoming link delay is added to the residence time in the correction field of the PTP message or an associated follow-up message.

Note PTP operates only in boundary clock mode. Cisco recommends deployment of a Grand Master Clock (10 MHz) upstream, with servers containing clocks requiring synchronization connected to the switch.

End-to-end transparent clock and peer-to-peer transparent clock modes are not supported.

Clock Modes

The IEEE 1588 standard specifies two clock modes for the PTP supporting devices to operate in: one-step and two-step.

One-Step Mode:

In one-step mode the clock synchronization messages include the time at which the master port sends the message. The ASIC adds the timestamp to the synchronization message as it leaves the port. The master port operating in one-step mode for Cisco Nexus 9508-FM-R and 9504-FM-R fabric modules and Cisco Nexus 9636C-R, 9636Q-R, and 9636C-RX line cards.

The slave port uses the timestamp that comes as part of the synchronization messages.

Two-Step Mode:

In two-step mode the time at which the synchronization message leaves the port is sent in a subsequent follow-up message. This is the default mode.

PTP Process

The PTP process consists of two phases: establishing the master-slave hierarchy and synchronizing the clocks.

Within a PTP domain, each port of an ordinary or boundary clock follows this process to determine its state:

- Examines the contents of all received announce messages (issued by ports in the master state)
- Compares the data sets of the foreign master (in the announce message) and the local clock for priority, clock class, accuracy, and so on
- Determines its own state as either master or slave

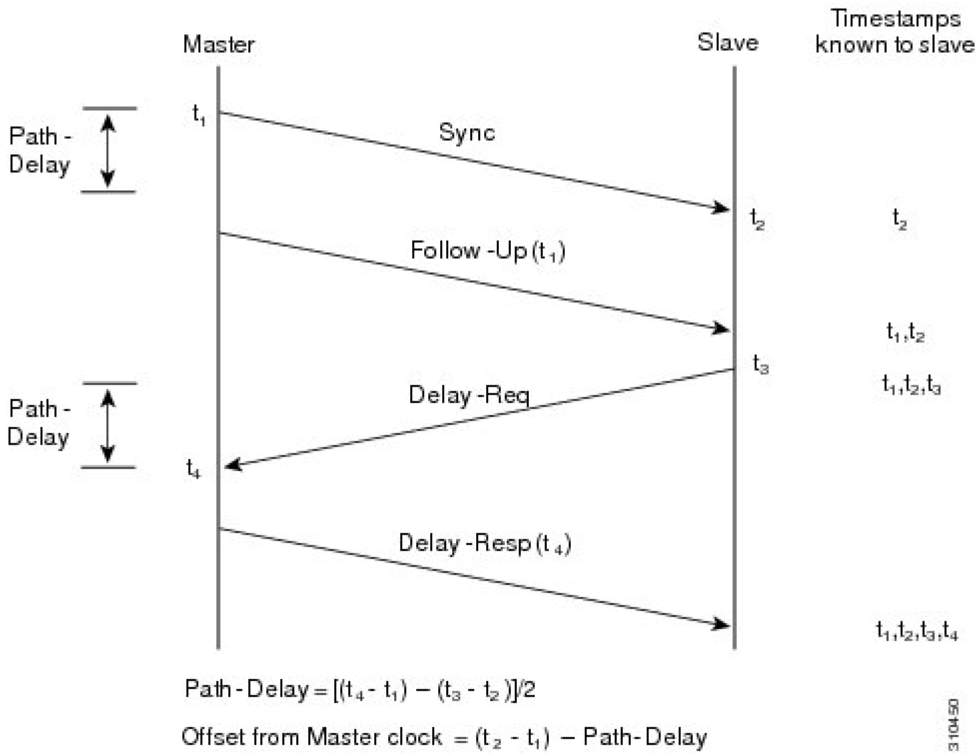
The ordinary and boundary clocks use **Sync**, **Delay_Req**, **Follow_Up**, **Delay_Resp** event messages to generate and communicate timing information.

These messages are sent in the following sequence:

1. The master sends a **Sync** message to the slave and notes the time, `t1` at which it was sent. For one-step **Sync** message carries the time when the message leaves the master and for two-step this time is sent in the subsequent **Follow-Up** event message.
2. The slave receives the **Sync** message and notes the time of reception, `t2`.
3. The master conveys to the slave the timestamp, `t1` by embedding the timestamp in a **Follow_Up** event message.
4. The slave sends a **Delay_Req** message to the master and notes the time, `t3` at which it was sent.
5. The master receives the **Delay_Req** message and notes the time of reception, `t4`.
6. The master conveys to the slave the timestamp, `t4` by embedding it in a **Delay_Resp** message.
7. After this sequence, the slave possesses all four timestamps. These timestamps can be used to compute the offset of the slave clock relative to the master, and the mean propagation time of messages between the two clocks.

   The following figure describes the event messages in the PTP process that generate and communicate timing information.

Figure 2: PTP Process



High Availability for PTP

Stateful restarts are not supported for PTP. 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.

Guidelines and Limitations for PTP

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

The following are the guidelines and limitations for Cisco Nexus 9000 series switches for PTP:

- For optimal PTP performance, it is recommended to use the latest SUP and LC FPGA versions for the release.
- PTP domain is limited to a single domain per network.
- PTP transport over User Datagram Protocol (UDP) is supported. Transport over Ethernet is not supported.
- PTP supports multicast communication. PTP also supports unicast communication and the unicast mode is optional.
- PTP supports boundary clock mode. End-to-end transparent clock and peer-to-peer transparent clock modes are not supported.
- It is recommended that the PTP device can either have multicast or unicast PTP mode configured, but not both multicast and unicast mode together.
- PTP can be enabled on port-channel member ports.
- All management messages received on a Slave port is forwarded on all PTP enabled ports. Handling management messages is not supported.
- One-step PTP is only supported on Cisco Nexus 9000-R series platform switches.
- Timestamp Tagging (TTAG) is supported on the following platform switches:
    - Cisco Nexus 9200 Series Platform Switches: beginning with Cisco NX-OS Release 7.0(3)I6(1)
    - Cisco Nexus 9364C: beginning with Cisco NX-OS Release 7.0(3)I7(2)
    - Cisco Nexus 9332C: beginning with Cisco NX-OS Release 9.2(3)
    - Cisco Nexus 9300-EX Series Platform Switches: beginning with Cisco NX-OS Release (7.0(3)I6(1)
    - Cisco Nexus 9300-FX platform switches: beginning with Cisco NX-OS Release 7.0(3)I7(3)
    - Cisco Nexus 9500 platform switches with -EX and -FX line cards
- To match PTP control packets using RACL, enable PIM on the L3 interface.
- When PTP is configured on a Cisco Nexus 9000 series switches, set the clock protocol to use PTP through the clock protocol ptp vdc 1 command. NTP cannot coexist with PTP on a Cisco Nexus 9000 series switch.
- PTP is supported for all Cisco Nexus 9000 series switches and 3164Q hardware except for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM).
- Beginning with Cisco NX-OS Release 9.2(3), PTP is supported on the N9K-C9504-FM-R.

- For Cisco Nexus 31108PC-V and 31108TC-V switches, PTP is not supported on ports running at 100G speed.
- Cisco Nexus 9000 series switches support mixed non-negotiated mode of operation on master PTP ports. That means when a slave client sends unicast delay request PTP packet, the Cisco Nexus 9000 responds with an unicast delay response packet. And, if the slave client sends multicast delay request PTP packet, the Cisco Nexus 9000 responds with a multicast delay response packet. For mixed non-negotiated mode to work, the source IP address used in the ptp source *IP address* configuration on the BC device must also be configured on any physical or logical interface of the BC device. The recommended best practice is to use the loopback interface of the device.
- Cisco Nexus 9000 series switches support mixed non-negotiated mode of operation on master PTP ports. Meaning that when a slave client sends unicast delay request PTP packet, the Cisco Nexus 9000 responds with an unicast delay response packet. And, if the slave client sends multicast delay request PTP packet, the Cisco Nexus 9000 responds with a multicast delay response packet.
- Beginning with Cisco NX-OS Release 9.2(1), Cisco Nexus 9732C-EX, 9736C-EX, and 97160YC-EX line cards support PTP offloading.
- PTP offload is not supported on the Cisco Nexus 9508 switch with an -R series line card.
- PTP is not supported on the Cisco Nexus 9504 switch with an -R series line card.
- PTP offload is not supported on the Cisco N3K-C36180YC-R and N3K-C3636C-R line cards.
- Beginning with Cisco NX-OS Release 9.2(1), Cisco Nexus 9636C-RX, 9636C-R, and 9636Q-R line cards support IEEE 802.1AS. IEEE 802.1AS is not supported on the Cisco Nexus 96136YC-R line card or the Cisco Nexus 9504 switch.
- PTP is not supported on the Cisco Nexus 96136YC-R line card.
- Cisco Nexus 93108TC-EX and 93180YC-EX switches support PTP mixed mode and unicast mode. The Cisco Nexus 9396 switch supports PTP mixed mode.
- PTP is supported with sync interval *-3* only on Cisco Nexus 9508-R line cards. Higher sync intervals are not supported.
- PTP unicast is supported only on the default vrf.
- PTP supports unicast communication on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.
- PTP unicast mode on the Layer2 SVI interfaces is not supported on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.
- PTP configuration with UC and MC on either side is not supported on Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- PTP is not supported on FEX interfaces.
- PTP-capable ports do not identify PTP packets and do not time-stamp or redirect those packets unless you enable PTP on those ports.
- Cisco Nexus 9500 platform switches with 9700-EX line cards support timestamp tagging.
- Each port can be individually configured with any of the supported PTP profiles. Different PTP profiles can coexist on an interface. Combination of the default of 1588 and SMPTE-2059-2 or AES67 profiles

is supported. However, combination of SMPTE-2059-2 and AES67 profiles is not supported on the same interface.

- Unicast negotiation is not supported for IPv4 unicast communication for all Nexus 9000 switches.

Default Settings for PTP

The following table lists the default settings for PTP parameters.

Table 3: Default PTP Parameters

Parameters	Default
PTP	Disabled
PTP version	2
PTP domain	0
PTP priority 1 value when advertising the clock	255
PTP priority 2 value when advertising the clock	255
PTP announce interval	1 log second
PTP announce timeout	3 announce intervals
PTP delay-request interval	• 0 log seconds • -1 log seconds for Cisco Nexus 3232C, 3264Q, and 9500 platform switches
PTP sync interval	• -2 log seconds • -3 log seconds for Cisco Nexus 3232C, 3264Q, and 9500 platform switches
PTP VLAN	gPTP supports only default vlan 1, and no other user configured VLANs.

Configuring PTP

Configuring PTP Globally

You can enable or disable PTP globally on a device. You can also configure various PTP clock parameters to help determine which clock in the network has the highest priority to be selected as the grandmaster.

Note You must always set the clock protocol PTP vdc1 for the local clock to be updated by the PTP protocol, irrespective of the one-step or the two-step mode. You can verify the configuration using the **show running-config clock_manager** command.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature ptp** **Example:** `switch(config)# feature ptp`	Enables or disables PTP on the device. **Note** Enabling PTP on the switch does not enable PTP on each interface.
Step 3	[**no**] **ptp device-type** [**generalized-ptp** | **boundary-clock**] **Example:** `switch(config)# ptp device-type generalized-ptp`	Configures the device type as gPTP or boundary clock. The **generalized-ptp** option is available only for Cisco Nexus 9508 switches with an -R series line card.
Step 4	[**no**] **ptp source** *ip-address* **Example:** `switch(config)# ptp source 10.10.10.1`	Configures the source IPv4 address for all the PTP packets in the multicast PTP mode.
Step 5	(Optional) [**no**] **ptp domain** *number* **Example:** `switch(config)# ptp domain 1`	Configures the domain number to use for this clock. PTP domains allow you to use multiple independent PTP clocking subdomains on a single network. The range for the *number* is from 0 to 127.
Step 6	(Optional) [**no**] **ptp priority1** *value* **Example:** `switch(config)# ptp priority1 1`	Configures the priority1 value to use when advertising this clock. This value overrides the default criteria (clock quality, clock class, and so on) for best master clock selection. Lower values take precedence. The range for the *value* is from 0 to 255. **Note** For the switch to synchronize with an external Grand Master clock, the local switch PTP priority value must be configured higher than that of external Grand Master Clock priority.

	Command or Action	Purpose
Step 7	(Optional) [**no**] **ptp priority2** *value* **Example:** `switch(config)# ptp priority2 1`	Configures the priority2 value to use when advertising this clock. This value is used to decide between two devices that are otherwise equally matched in the default criteria. For example, you can use the priority2 value to give a specific switch priority over other identical switches. The range for the *value* is from 0 to 255. **Note** For the switch to synchronize with an external Grand Master clock, the local switch PTP priority value must be configured higher than that of external Grand Master Clock priority.

Step 8	[**no**] **ptp management**
	Example:
	`switch(config)# ptp management`
	`switch(config-ptp-profile)#`
		Configures support for PTP management packets. This command is enabled by default.
		no: Disables support for management packets.

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 PTP on an Interface

After you globally enable PTP, it is not enabled on all supported interfaces by default. You must enable PTP interfaces individually.

Before you begin

Make sure that you have globally enabled PTP on the switch and configured the source IP address for PTP communication.

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	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling PTP and enters the interface configuration mode.

Step 3	[**no**] **ptp** **Example:** `switch(config-if)# ptp`	Enables or disables PTP on an interface.

Step 4	(Optional) [**no**] **ptp master** *ipv6-address*
	Example:
	`switch(config-if)# ptp master 10.10.10.1`
	`switch(config-if)#`
		Sets the IP address of the master clock when the PTP role of the interface is set to "slave".
		This option is only available for PTP Telecom Profile on the Cisco Nexus 93180YC-FX3S switch.
		Note This command is supported beginning with Cisco NX-OS Release 10.2(1)F.

Step 5	(Optional) **ptp transport ipv4 ucast master** **Example:** `switch(config)# ptp transport ipv4 ucast master`	Enables PTP on the interface in an unicast mode (PTP master use).
Step 6	(Optional) **ptp transport ipv4 ucast slave** **Example:** `switch(config)# ptp transport ipv4 ucast slave`	Enables PTP on the interface in an unicast mode (PTP slave use).
Step 7	(Optional) **ptp ucast-source** *ipv4-address* [**vrf** *vrf-name*] **Example:** `switch(config)# ptp ucast-source 10.1.1.40`	Configures the source IP address for unicast messages. *ipv4-address*: The IPv4 address of the unicast source. **vrf** *vrf-name*: The name of the VRF used for hello messages.
Step 8	(Optional) [**no**] **ptp announce** {**interval** *log-seconds* | **timeout** *count*} **Example:** `switch(config-if)# ptp announce interval 3`	Configures the interval between PTP announce messages on an interface or the number of PTP intervals before a timeout occurs on an interface. The range for the PTP announcement interval is from 0 to 4 log seconds, and the range for the interval timeout is from 2 to 4 intervals.
Step 9	(Optional) [**no**] **ptp announce interval** [**aes67** | **smpte-2059**] *log-seconds* **Example:**	Configures the interval between PTP announce messages on an interface.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td>

```
switch(config-if)# ptp announce interval
 aes67 3
```

</td><td>

Table 4: PTP Announcement Interval Range and Default Values

<table>
<tr><th>Option</th><th>Range</th><th>Default Value</th></tr>
<tr><td>aes67-2015</td><td>0 to 4 log seconds</td><td>1 log second</td></tr>
<tr><td>smpte-2059-2</td><td>–3 to 1 log seconds</td><td>1 log second</td></tr>
<tr><td>Without the aes67 or smpte-2059 option</td><td>0 to 4 log seconds</td><td>1 log second</td></tr>
</table>

</td></tr>
<tr><td>Step 10</td><td>

(Optional) [**no**] **ptp delay-request minimum interval** *log-seconds*

Example:

```
switch(config-if)# ptp delay-request
minimum interval -1
```

</td><td>

Configures the minimum interval allowed between PTP delay messages when the port is in the master state.

The range is from log(–1) to log(6) seconds, where log(–1) = 2 frames every second.

</td></tr>
<tr><td>Step 11</td><td>

(Optional) [**no**] **ptp delay-request minimum interval** [**aes67-2015** | **smpte-2059-2**] *log-seconds*

Example:

```
switch(config-if)# ptp delay-request
minimum interval aes67-2015-1
```

</td><td>

Configures the minimum interval allowed between PTP delay messages when the port is in the master state.

Table 5: PTP Delay-Request Minimum Interval Range and Default Values

<table>
<tr><th>Option</th><th>Range</th><th>Default Value</th></tr>
<tr><td>aes67-2015</td><td>–4 to 5 log seconds</td><td>0 log seconds</td></tr>
<tr><td>smpte-2059-2</td><td>–4 to 5 log seconds</td><td>0 log seconds</td></tr>
<tr><td>Without the aes67-2015 or smpte-2059-2 option</td><td>–1 to 6 log seconds (where –1 = 2 frames every second)</td><td>0 log seconds</td></tr>
</table>

</td></tr>
<tr><td>Step 12</td><td>

(Optional) [**no**] **ptp sync interval** *log-seconds*

Example:

```
switch(config-if)# ptp sync interval 1
```

</td><td>

Configures the interval between PTP synchronization messages on an interface.

The range is from log(–3) to log(1) seconds. For the media-related profile information, see the Cisco NX-OS IP Fabric for Medial Solution Guide when configuring PTP for media.

</td></tr>
</table>

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 13</td><td>(Optional) [no] ptp sync interval [aes67-2015 | smpte-2059-2] log-seconds
Example:
<code>switch(config-if)# ptp sync interval aes67 1</code></td><td>Configures the interval between PTP synchronization messages on an interface.
Table 6: PTP Synchronization Interval Range and Default Values
<table>
<tr><th>Option</th><th>Range</th><th>Default Value</th></tr>
<tr><td>aes67-2015</td><td>–4 to 1 log seconds</td><td>–2 log seconds</td></tr>
<tr><td>smpte-2059-2</td><td>–4 to –1 log seconds</td><td>–2 log seconds</td></tr>
<tr><td>Without the aes67-2015 or smpte-2059-2 option</td><td>–3 to 1 log seconds</td><td>–2 log seconds</td></tr>
</table></td></tr>
<tr><td>Step 14</td><td>(Optional) [no] ptp vlan vlan-id
Example:
<code>switch(config-if)# ptp vlan 1</code></td><td>Specifies the VLAN for the interface where PTP is being enabled. You can only enable PTP on one VLAN on an interface.
The range is from 1 to 4094.</td></tr>
<tr><td>Step 15</td><td>(Optional) show ptp brief
Example:
<code>switch(config-if)# show ptp brief</code></td><td>Displays the PTP status.</td></tr>
<tr><td>Step 16</td><td>(Optional) show ptp port interface interface slot/port
Example:
<code>switch(config-if)# show ptp port interface ethernet 2/1</code></td><td>Displays the status of the PTP port.</td></tr>
<tr><td>Step 17</td><td>(Optional) copy running-config startup-config
Example:
<code>switch(config-if)# copy running-config startup-config</code></td><td>Copies the running configuration to the startup configuration.</td></tr>
</table>

Configuring PTP in Unicast Mode

Configuring Unicast Mode

Traditional PTP messages are delivered to the nodes that are capable of receiving PTP multicast messages. (For example, **announce**, **sync**, **delay_req**, **delay_resp** and **follow_up**). In Unicast mode, all PTP messages are delivered only to a particular PTP node. Multicast address is not used. In unicast mode, you can configure master/slave role and assign corresponding peer slave/master IP addresses.

Up to 8 master IPs can be configured for a slave unicast port and 64 slave IPs can be configured for a master port with a maximum 256 slave IP total for all ports. The following commands are used to configure the unicast slave IPs and unicast master IPs. Unicast packets are only sent to and received from these IPs. Packets received from other IPs are ignored.

```
switch(config-if)# ptp transport ipv4 ucast master
switch(config-if-ptp-master)# slave ipv4 10.10.10.2

switch(config-if)# ptp transport ipv4 ucast slave
switch(config-if-ptp-slave)# master ipv4 10.10.10.1
```

Assigning Master Role

Complete the following steps to assign a master role:

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 2/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling PTP and enters the interface configuration mode.

Step 3	[**no**] **ptp transport ipv4 ucast master**
	Example:
	`switch(config-if)# ptp transport ipv4 ucast master`
	`switch(config-if-ptp-master)#`
		Enables PTP master on a particular port (Layer 3 interface). In the master sub-mode, you can enter the slave IPv4 addresses.

Step 4	**slave ipv4** *<IP_address>*
	Example:
	`switch-1(config)# interface ethernet 1/1`
	`switch-1(config-if)# ptp transport ipv4 ucast master`
	`switch-1(config-if-ptp-master)# slave ipv4 1.2.3.1`
	`switch-1(config-if-ptp-master)# slave ipv4 1.2.3.2`
	`switch-1(config-if-ptp-master)# slave ipv4 1.2.3.3`
	`switch-1(config-if-ptp-master)# slave ipv4 1.2.3.4`
	`switch-1(config-if-ptp-master)#`
		Enters the slave IPv4 addresses. Maximum of 64 IP addresses are allowed per master, but this number varies and it depends on the sync interval configuration. The master sends announce, sync, follow-up, and delay_resp only to these slave addresses. You must make sure that the slave IP is reachable.

Assigning Slave Role

Complete the following steps to assign a slave role:

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 2/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling PTP and enters the interface configuration mode.

Step 3	[**no**] **ptp transport ipv4 ucast slave**
	Example:
	`switch(config-if)# ptp transport ipv4 ucast slave`
	`switch(config-if-ptp-slave)#`
		Enables PTP slave on a particular port (Layer 3 interface). In the slave sub-mode, you can enter the master IPv4 addresses.

Step 4	**master ipv4** *<IP_address>*
	Example:
	`switch-1(config)# interface ethernet 1/1`
	`switch-1(config-if)# ptp transport ipv4 ucast slave`
	`switch-1(config-if-ptp-slave)# master ipv4 4.4.4.1`
	`switch-1(config-if-ptp-slave)# master ipv4 4.4.4.2`
	`switch-1(config-if-ptp-slave)# master ipv4 4.4.4.3`
		Enters the master IPv4 addresses.

Configuring Unicast Source Address

Note For all releases up to, and including Cisco NX-OS Release 9.3(4), if the PTP configuration on the interface is changed from unicast to multicast or unicast slave to unicast master, you must reconfigure the unicast source address.

Complete the following steps to configure unicast source address:

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	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling PTP and enters the interface configuration mode.

Step 3	[**no**] **ptp ucast-source** *ipv4-address*
	Example:
	`switch(config-if)# ptp ucast-source 10.10.10.20`
	`switch(config-if)#`
		Configure PTP source address per interface level. This IP address is used only for unicast PTP messages. The PTP unicast source IP address must be reachable.

Timestamp Tagging

The timestamp tagging feature provides precision time information to track in real time when packets arrive at remote devices. Packets are truncated and timestamped using PTP with nanosecond accuracy. Using the TAP aggregation functionality on the switch, along with the Cisco Nexus Data Broker, you can copy the network traffic using SPAN, filter and timestamp the traffic, and send it for recording and analysis.

Configuring Timestamp Tagging

Note Configuring timestamp tagging is not supported on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.

Note When you use the ttag feature in a VXLAN EVPN multisite deployment, make sure that the ttag is stripped (**ttag-strip**) on BGW's DCI interfaces that connect to the cloud. To elaborate, if the ttag is attached to non-Nexus 9000 devices that do not support ether-type 0x8905, stripping of ttag is required. However, BGW back-to-back model of DCI does not require ttag stripping.

Before you begin

Make sure that you have globally enabled PTP offloading.

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	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters interface configuration mode for the specified interface.

Step 3	[**no**] **ttag** **Example:** `switch(config-if)# ttag`	Configures timestamp tagging on the Layer 2 or Layer 3 egress interface.

Configuring the TTAG Marker Packets and Time Interval

The ttag timestamp field attaches a 48-bit timestamp on the marker packet. This 48-bit timestamp is not a human familiar ASCII based timestamp. To make this 48-bit timestamp human readable, the ttag marker packet can be used to provide additional information to decode the 48-bit timestamp information.

Field	**Position (byte:bit)**	**Length**	**Definition**
Magic		16	By default, this field displays A6A6. This enables to identify ttag-marker packets on the packet stream.
Version		8	Version number. The default version is 1.
Granularity		16	This field represents the granularity of the 48-bit timestamp size. By default, the value is 04, which is 100 picoseconds or 0.1. nanoseconds.
UTc_offset		8	The utc_offset between the ASIC and the UTC clocks. The default value is 0.
Timestamp_hi		32	The high 16-bit of 48- bit ASIC hardware timestamp.
Timestamp_lo		32	The low 32-bit of 48- bit ASIC hardware timestamp.
UTC sec		32	The seconds part of UTC timestamp from the CPU clock of the Cisco Nexus 9000 Series switch.
UTC nsec		32	The nanoseconds part of UTC timestamp from the CPU clock of the Cisco Nexus 9000 Series switch.
Reserved		32	Reserved for future use.

Signature		32	The default value is 0xA5A5A5A5. This allows a forward search of marker packet and provide references to the UTC timestamp, so the client software can use that reference UTC to recover the 32-bit hardware timestamp in each packet header.
Pad		8	This is align byte to convert the ttag-marker align to 4 byte boundary.

Before you begin

Make sure that you have globally enabled PTP offloading.

Procedure

	Command or Action	Purpose

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

Step 2	**ttag-marker-interval** *seconds* **Example:** `switch(config-if)# ttag-marker-interval 90`	Configures the seconds that a switch will take to send a ttag-marker packet to the outgoing ports. This is a global setting to the switch. By default, it sends a ttag-marker packet every 60 seconds. The range for seconds is from 1 to 25200.

Step 3	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters interface configuration mode for the specified interface.

Step 4	[**no**] **ttag-marker enable** **Example:** `switch(config-if)# ttag-marker enable`	Sends the ttag-marker packets to the outgoing port.
Step 5	**ttag-strip** **Example:** `switch(config-if)# ttag-strip`	Removes TTAG from egress packets on the interface.

Configuring a PTP Interface to Stay in a Master State

This procedure describes how to prevent an endpoint from causing a port to transition to a slave state.

Before you begin

- Make sure that you have globally enabled PTP on the switch and configured the source IP address for PTP communication.
- After you globally enable PTP, it is not enabled on all supported interfaces by default. You must enable PTP interfaces individually.

Procedure

	Command or Action	Purpose
Step 1	switch # **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **interface ethernet** *slot/port*	Specifies the interface on which you are enabling PTP and enters interface configuration mode.
Step 3	switch(config-if) # **ptp**	Enables or disables PTP on an interface.
Step 4	switch(config-if) # **ptp multicast master-only**	Configures the port to maintain the master state.

Example

This example shows how to configure PTP on an interface and configure the interface to maintain the Master state:

```
switch(config)# show ptp brief

PTP port status
----------------------------------
Port                  State
----------------  ----------------
Eth1/1              Slave
switch(config)# interface ethernet 1/1
switch(config-if)# ptp multicast master-only
2001 Jan  7 07:50:03 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_GM_CHANGE: Grandmaster clock has changed
 from 60:73:5c:ff:fe:62:a1:41 to 58:97:bd:ff:fe:0d:54:01 for the PTP protocol
2001 Jan  7 07:50:03 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_STATE_CHANGE: Interface Eth1/1 change from
 PTP_BMC_STATE_SLAVE to PTP_BMC_STATE_PRE_MASTER
2001 Jan  7 07:50:03 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_TIMESYNC_LOST: Lost sync with  master clock
2001 Jan  7 07:50:07 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_STATE_CHANGE: Interface Eth1/1 change from
 PTP_BMC_STATE_PRE_MASTER to PTP_BMC_STATE_MASTER
```

Verifying the PTP Configuration

Use one of the following commands to verify the configuration:

Table 7: PTP Show Commands

Command	Purpose
show ptp brief	Displays the PTP status.

Command	Purpose
show ptp clock	Displays the properties of the local clock, including clock identity.
show ptp clock foreign-masters-record	Displays the state of foreign masters known to the PTP process. For each foreign master, the output displays the clock identity, basic clock properties, and whether the clock is being used as a grandmaster.
show ptp corrections	Displays the last few PTP corrections.
show ptp counters [**all** | **interface ethernet** *slot/port*]	Displays the PTP packet counters for all interfaces or for a specified interface.
show ptp parent	Displays the properties of the PTP parent.
show ptp port interface ethernet *slot/port*	Displays the status of the PTP port on the switch.
show ptp time-property	Displays the PTP clock properties.
show ttag brief	Displays the status of the timestamp tagging configuration. **Note** Timestamp tagging configuration is not supported on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.
show running-config ptp [**all**]	Displays the running configuration for PTP.
clear ptp counters [**all** | **interface ethernet** *slot/port*]	Clears all PTP messages that are received and transmitted on a specific interface or on all interfaces that has PTP enabled.
clear ptp corrections	Clears the history of the PTP corrections.

Configuration Examples for PTP

This example shows how to configure PTP globally on the device, specify the source IP address for PTP communications, and configure a preference level for the clock:

```
switch# configure terminal
switch(config)# feature ptp
switch(config)# ptp source 10.10.10.1
switch(config)# ptp priority1 1
switch(config)# ptp priority2 1
switch(config)# show ptp brief
PTP port status
-----------------------
Port State
------- --------------
switch(config)# show ptp clock
PTP Device Type: Boundary clock
```

```
Clock Identity : 0:22:55:ff:ff:79:a4:c1
Clock Domain: 0
Number of PTP ports: 0
Priority1 : 1
Priority2 : 1
Clock Quality:
  Class : 248
  Accuracy : 254
  Offset (log variance) : 65535
Offset From Master : 0
Mean Path Delay : 0
Steps removed : 0
Local clock time:Mon Dec 22 14:13:24 2014
```

This example shows how to configure PTP on an interface and configure the intervals for the announce, delay-request, and synchronization messages:

```
switch# configure terminal
switch(config)# interface Ethernet 1/1
switch(config-if)# ptp
switch(config-if)# ptp announce interval 3
switch(config-if)# ptp announce timeout 2
switch(config-if)# ptp delay-request minimum interval smpte-2059-2 -3
switch(config-if)# ptp sync interval smpte-2059-2 -3
switch(config-if)# no shutdown
switch(config-if)# show ptp brief
PTP port status
-----------------------
Port State
------- --------------
Eth2/1 Master
switch(config-if)# show ptp port interface ethernet 2/1
PTP Port Dataset: Eth2/1
Port identity: clock identity: 0:22:55:ff:ff:79:a4:c1
Port identity: port number: 1028
PTP version: 2
Port state: Master
Delay request interval(log mean): 4
Announce receipt time out: 2
Peer mean path delay: 0
Announce interval(log mean): 3
Sync interval(log mean): 1
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
```

This example shows how to configure master/slave role and assign corresponding peer slave/master IP addresses.

```
For Cisco NX-OS Release 9.3(4) and earlier:
switch-1(config)# interface ethernet 1/1
switch-1(config-if)# ptp transport ipv4 ucast master
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.1
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.2
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.3
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.4
switch-1(config-if-ptp-master)#

switch-1(config-if)# ptp transport ipv4 ucast slave
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.1
```

```
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.2
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.3


switch-1(config-if-ptp-slave)# ptp ucast-source 9.9.9.9


switch-1(config-if)# sh running-config ptp

!Command: show running-config ptp
!Time: Tue Feb  7 17:37:09 2017

version 7.0(3)I4(6)
feature ptp

ptp source 1.1.1.1


interface Ethernet1/1
  ptp transport ipv4 ucast master
    slave ipv4 1.2.3.1
    slave ipv4 1.2.3.2
    slave ipv4 1.2.3.3
    slave ipv4 1.2.3.4

interface Ethernet1/2
  ptp transport ipv4 ucast slave
    master ipv4 4.4.4.1
    master ipv4 4.4.4.2
    master ipv4 4.4.4.3
  ptp ucast-source 9.9.9.9

switch-1(config-if)#
```

Additional References

Related Documents

Related Topic	Document Title
1588 IEEE	1588 IEEE standards

CHAPTER 5

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.
- Before configuring a name based NTP server (FQDN) in a non-default VRF, you must configure a DNS server under that specific VRF. If you configure the DNS server from the global configuration mode using **use-vrf** option, then that name based NTP server configuration will not be added to the running configuration. If you attempted to configure NTP server using this method, you must remove the NTP configuration using the **no** version of the command, add the DNS server under that VRF, and then add name based NTP server to the VRF. The configured DNS server must be reachable and must return the correct IP for the FQDN of the NTP server when queried.
- We recommend that you configure 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 that is 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, we recommend that you 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.
- 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, we recommend 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.

Note Make sure that you specify **ntp authenticate** before enabling any of the preceding commands. Failure to do so will allow your device to synchronize with any device that sends one of the preceding packet types, including malicious attacker-controlled devices.

- If you specify the **ntp authenticate** command, 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 that are 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 that are configured via the **ntp server** and **ntp peer** configuration commands. To authenticate the **ntp server** and **ntp peer** associations, specify the **key** keyword.
- A maximum of four IP ACLs can be configured for a single NTP access group. IPv4 and IPv6 ACLs are supported.

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 access group match all	Disabled
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:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
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 *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

	Command or Action	Purpose
		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:** `switch(config)# show ntp peers`	Displays the configured server and 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.

	Command or Action	Purpose
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 15 alphanumeric characters.
Step 3	**ntp server** *ip-address* **key** *key-id* **Example:** `switch(config)# ntp server 192.0.2.1 key 1001`	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. 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.

	Command or Action	Purpose
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.

Beginning with Cisco NX-OS Release 7.0(3)I7(3), access groups are evaluated in the following method:

- 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).

The mapping of the access group to the type of packet is as follows:

- 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.

	Command or Action	Purpose

Step 2	[**no**] **ntp access-group match-all** | {{**peer** | **serve** | **serve-only** | **query-only** }*access-list-name*}
	
	Example:
	`switch(config)# ntp access-group match-all`
	`switch(config)# ntp access-group peer peer-acl`
	`switch(config)# ntp access-group serve serve-acl`
		Creates or removes an access group to control NTP access and applies a basic IP access list.
		
		ACL processing stops and does not continue to the next access group option if NTP matches a deny ACL rule in a configured peer.
		
		• The **peer** keyword enables the device to receive time requests and NTP control queries and to synchronize itself to the servers specified in the access list.
		
		• The **serve** keyword enables the device to receive time requests and NTP control queries from the servers specified in the access list but not to synchronize itself to the specified servers.
		
		• The **serve-only** keyword enables the device to receive only time requests from servers specified in the access list.
		
		• The **query-only** keyword enables the device to receive only NTP control queries from the servers specified in the access list.
		
		• The **match-all** keyword enables the access group options to be scanned in the following order, from least restrictive to most restrictive: peer, serve, serve-only, query-only. If the incoming packet does not match the ACL in the peer access group, it goes to the serve access group to be processed. If the packet does not match the ACL in the serve access group, it goes to the serve-only access group, and so on.
		
		Note The **match-all** keyword is available beginning with Cisco NX-OS Release 7.0(3)I6(1) and is supported on Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches.
		
		• The *access-list-name* variable is the name of the NTP access group. The name can be an alphanumeric string up to 64 characters, including special characters.

	Command or Action	Purpose
Step 3	(Optional) **show ntp access-groups** **Example:** `switch(config)# show ntp access-groups`	Displays the NTP access group 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 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.

	Command or Action	Purpose
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.

Command	Purpose
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 https://cisco.github.io/cisco-mibs/supportlists/nexus9000/ Nexus9000MIBSupportList.html

CHAPTER 6

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.
- Beginning with Cisco NX-OS Release 10.4(2)F, CDP is supported on Cisco Nexus 9232E-B1 platform switches.

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.

	Command or Action	Purpose
Step 3	(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 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 7

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 8: 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.

Secure Syslog Servers

Beginning with Cisco NX-OS Release 9.2(1), you can configure the syslog server with support for a secure TLS transport connectivity to remote logging servers. Additionally, you can enforce the NX-OS switches (client) identity via the mutual authentication configuration. For NX-OS switches, this feature supports TLSv1.1 and TLSv1.2.

The Secure syslog server feature uses the TCP/TLS transport and security protocols to provide device authentication and encryption. This feature enables a Cisco NX-OS device (acting as a client) to make a secure, encrypted outbound connection to remote syslog servers (acting as a server) supporting secure connectivity for logging. With authentication and encryption, this feature allows for a secure communication over an insecure network.

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.

- Beginning with Cisco NX-OS Release 9.2(1), you can configure the syslog server with support for a secure TLS transport connectivity to remote logging servers. This feature supports TLS v1.1 and TLS v1.2.
- For the secure syslog server(s) to be reachable over an in-band (nonmanagement) interface, the CoPP profile may need tweaks. Especially when multiple logging servers are configured and when many syslogs are generated in a short time (such as, boot up and config application).
- 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 9: Default System Message Logging Parameters

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

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

	Command or Action	Purpose
		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. 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.

Configuring the Origin ID for Syslog Messages

You can configure Cisco NX-OS to append the hostname, an IP address, or a text string to syslog messages that are sent to remote syslog servers.

Procedure

	Command or Action	Purpose

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

Step 2	Required: **logging origin-id** {**hostname** | **ip** *ip-address* | **string** *text-string*} **Example:** `switch(config)# logging origin-id string n9k-switch-abc`	Specifies the hostname, IP address, or text string to be appended to syslog messages that are sent to remote syslog servers.

Step 3	(Optional) **show logging origin-id**
	Example:
	`switch(config)# show logging origin-id`
	`Logging origin_id : enabled (string: n9k-switch-abc)`
		Displays the configured hostname, IP address, or text string that is appended to syslog messages that are sent to remote syslog servers.

Step 4	(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:

	Command or Action	Purpose
		• 0 – emergency • 1 – alert • 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:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

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

	Command or Action	Purpose
		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 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.

	Command or Action	Purpose
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.
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::3 5 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. The **use-vrf** *vrf-name* keyword identifies the default or management values for the VRF name. The default VRF is the management VRF, by default. However, the **show-running** command will not list the default VRF. Severity levels range from 0 to 7:
		• 0 – emergency

	Command or Action	Purpose
		• 1 – alert • 2 – critical • 3 – error • 4 – warning • 5 – notification • 6 – informational • 7 – debugging 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 Secure Syslog Servers

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	[**no**] **logging server** *host* [*severity-level* [**port** *port-number*]][**secure**[**trustpoint client-identity** *trustpoint-name*]][**use-vrf** *vrf-name*]]
	Example:
	`switch(config)# logging server 192.0.2.253 secure`
	Example:
	`switch(config)# logging server 2001::3 5 secure trustpoint client-identity myCA use-vrf red`
		Configures a syslog server at the specified hostname or IPv4 or IPv6 address. Optionally, you can enforce a mutual authentication by installing the client identity certificate that is signed by any CA and using the trustpoint client-identity option.
		The default destination port for a secure TLS connection is 6514.

Step 3	(Optional) **logging source-interface** *interface name* **Example:** `switch(config)# logging source-interface lo0`	Enables a source interface for the remote syslog server.
Step 4	(Optional) **show logging server** **Example:** `switch(config)# show logging server`	Displays the syslog server configuration. If the secure option is configured, the output will have an entry with the transport information. By default, the transport is UDP if the secure option is not configured.
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 CA Certificate

For the secure syslog feature support, the remote servers must be authenticated via a trustpoint configuration.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **crypto ca trustpoint** *trustpoint-name*
	Example:
	`switch(config)# crypto ca trustpoint winca`
	`switch(config-trustpoint)#`
		Configures a trustpoint.
		Note You must configure the ip domain-name before the trustpoint configuration.

	Command or Action	Purpose
Step 3	Required: **crypto ca authenticate** *trustpoint-name* **Example:** `switch(config-trustpoint)# crypto ca authenticate winca`	Configures a CA certificate for the trustpoint.
Step 4	(Optional) **show crypto ca certificate** **Example:** `switch(config)# show crypto ca certificates`	Displays the configured certificate/chain and the associated trustpoint.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration so that the trustpoint is persistent across the reload of the device.

Enrolling the CA Certificate

For mutual authentication, where the remote server wants the NX-OS switch (the client) to identify, that the peer authentication is mandatory, this is an additional configuration to enroll the certificate on the switch.

Procedure

	Command or Action	Purpose

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

Step 2	Required: **crypto key generate rsa label** *key name* **exportable modules 2048** **Example:** `switch(config-trustpoint)# crypto key generate rsa label myKey exportable modulus 2048`	Configure an RSA key pair. By default, the Cisco NX-OS software generates an RSA key using 1024 bits.

Step 3	[**no**] **crypto ca trustpoint** *trustpoint-name*
	Example:
	`switch(config)# crypto ca trustpoint myCA`
	`switch(config-trustpoint)#`
		Configures a trustpoint.
		Note You must configure the ip domain-name before the trustpoint configuration.

Step 4	Required: **rsakeypair** *key-name* **Example:** `switch(config-trustpoint)# rsakeypair myKey`	Associates the keypair generated to the trustpoint CA.

	Command or Action	Purpose
Step 5	**crypto ca trustpoint** *trustpoint-name* **Example:** `switch(config)# crypto ca authenticate myCA`	Configures a CA certificate for the trustpoint.
Step 6	[**no**] **crypto ca enroll** *trustpoint-name* **Example:** `switch(config)# crypto ca enroll myCA`	Generate an identity certificate of the switch to enroll it to a CA.
Step 7	**crypto ca import** *trustpoint-name* **certificate** **Example:** `switch(config-trustpoint)# crypto ca import myCA certificate`	Imports the identity certificate signed by the CA to the switch.
Step 8	(Optional) **show crypto ca certificates** **Example:** `switch# show crypto ca certificates`	Displays the configured certificate or chain and the associated trustpoint.
Step 9	Required: **copy running-config startup-config** **Example:** `switch# 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 10: 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 origin-id	Displays the configured hostname, IP address, or text string that is appended to syslog messages that are sent to remote syslog servers.
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 8

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 11: 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 and Syslogs use different severity levels (see the following table). 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 12: 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.

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.
- Callhome does not support specifying a source interface using the **ip http source-interface** command.

Default Settings for Smart Call Home

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

Table 13: 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 103
2. Creating a Destination Profile, on page 105
3. Associating an Alert Group with a Destination Profile, on page 108
4. (Optional) Adding Show Commands to an Alert Group, on page 109
5. Enabling or Disabling Smart Call Home, on page 115
6. (Optional) Testing the Smart Call Home Configuration, on page 116

Configuring Contact Information

You must configure the email, phone, and street address information for Smart Call Home. You can optionally configure the contact 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&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 https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

CHAPTER 9

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:

- Configuration for only one service access point (SAP) can be performed using one session.
- Configuration sessions are not persistent across reloads.
- 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 than one configuration session or configuration terminal mode. Parallel configurations (for example, one configuration session and one configuration terminal) could cause validation or verification failures in the configuration session.
- If an interface reloads while you are configuring it in a configuration session, Session Manager can accept the commands even if the interface is not present in the device.

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*] **Example:** `switch(config-s)# show configuration session myACLs`	Displays the contents of the session.
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:**	Specifies the direction of traffic the access group is applied to.

	Command or Action	Purpose
	`switch(config-s-if)# ip access-group acl1 in`	
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 10

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.
- The scheduler accepts `start_time` in the past for any schedule with a repeat option in the time command under schedule mode configuration. It then throws a warning that the entered start time is in the past. The `start_time` of any schedule will always remain the same as it was in the beginning, across reboot, and even after re-applying the previous saved configuration.

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 11

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

Trap Type	Description
license	: notify-license-expiry
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 14: 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:

- Commands configured using SNMP SET should be deleted using SNMP SET only. Commands configured using Command Line Interface(CLI) or NX-API should be deleted using CLI or NX-API only.
- 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: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html
- Cisco NX-OS does not support the SNMPv3 noAuthNoPriv security level.
- Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches support the configuration of the SNMP local engine ID.
- For a nondisruptive downgrade path to an earlier release, if a local engine ID has been configured, then you must unconfigure the local engine ID, and then reconfigure the SNMP users and the community strings.
- 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.
- There must be at least one running BGP instance to have full, proper functional behavior of SNMP traps. Configure a BGP routing instance before configuring any snmp-server traps related commands.

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.
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:**	Copies the running configuration to the startup configuration.

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

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.

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

You can assign an access control list (ACL) to an SNMPv2 community or SNMPv3 user 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* | **use-ipv6acl** *acl-name*] **Example:** `switch(config)# snmp-server community public use-ipv4acl myacl`	Assigns an IPv4 or IPv6 ACL to an SNMPv2 community to filter SNMP requests. **Note** IPv6 ACLs are supported for SNMPv2 communities beginning with Cisco NX-OS Release 7.0(3)I4(1). **Note** In releases prior to Cisco NX-OS Release 7.0(3)I4(1), this CLI command includes **use-acl** rather than **use-ipv4acl**.
Step 3	**snmp-server user** *username* [**use-ipv4acl** *acl-name* | **use-ipv6acl** *acl-name*] **Example:** `switch(config)# snmp-server user user1 use-ipv4acl myacl`	Assigns an IPv4 or IPv6 ACL to an SNMPv3 user to filter SNMP requests. **Note** IPv6 ACLs are supported for SNMPv3 users beginning with Cisco NX-OS Release 7.0(3)I4(1). **Note** In releases prior to Cisco NX-OS Release 7.0(3)I4(1), this CLI command includes **use-acl** rather than **use-ipv4acl**.
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 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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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. **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 15: 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**
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp** [*tag*]
CISCO-ERR-DISABLE-MIB	**snmp-server enable traps link cerrDisableInterfaceEventRev1**

MIB	Related Commands
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-MAC-NOTIFICATION-MIB	**snmp-server enable trap link cmn-mac-move-notification**
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.

Managing the SNMP Server Counter Cache Update Timer

You can modify how long, in seconds Cisco NX-OS holds the cache port state.

Procedure

	Command or Action	Purpose

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

Step 2	**snmp-server counter cache timeout** *seconds* **Example:** `switch(config)# snmp-server counter cache timeout 1200`	Defines how long in seconds, the port states are held in the local cache. The counter cache is enabled by default, and the default cache timeout value is 10 seconds. When disabled, the default cache timeout value is 50 seconds. The range is 1-3600. **Note** For end of row (EoR) switching - The range is from 10 to 3600.
Step 3	(Optional) **show running-config snmp all |i cac** **Example:** `switch(config)# copy running-config snmp all | i cac`	Displays the configured SNMP-server counter cache update timeout value.
Step 4	**no snmp-server counter cache enable** **Example:** `switch(config)# no snmp-server counter cache enable`	Disables the counter cache update. **Note** When the counter cache update is disabled, the value set in the timeout parameter determines length of time the port states are held the counter cache.

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.

	Command or Action	Purpose
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.

Configuring the SNMP Local Engine ID

Beginning with Cisco NX-OS Release 7.0(3)I6(1), you can configure the engine ID on a local device.

Note After you configure the SNMP local engine ID, you must reconfigure all SNMP users, any host configured with the V3 users, and the community strings. Beginning with Cisco NX-OS Release 7.0(3)I7(1), you need to reconfigure only the SNMP users and community strings.

Procedure

	Command or Action	Purpose

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

Step 2	**snmp-server engineID local** *engineid-string* **Example:** `switch(config)# snmp-server engineID local AA:BB:CC:1A:2C:10`	Changes the SNMP engine ID of the local device. The local engine ID should be configured as a list of colon-specified hexadecimal octets, where there are even number of hexadecimal characters that range from 10 to 64 and every two hexadecimal characters are separated by a colon. For example, 80:00:02:b8:04:61:62:63.
Step 3	**show snmp engineID** **Example:** `switch(config)# show snmp engineID`	Displays the identification of the configured SNMP engine.
Step 4	[**no**] **snmp-server engineID local** *engineid-string* **Example:**	Disables the local engine ID and the default auto-generated engine ID is configured.

	Command or Action	Purpose
	`switch(config)# no snmp-server engineID local AA:BB:CC:1A:2C:10`	
Step 5	Required: **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.
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.

Command	Purpose
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	Title
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 following https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

CHAPTER 12

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 https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

CHAPTER 13

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 16: Bootup Diagnostics

Diagnostic	Description
OBFL	Verifies the integrity of the onboard failure logging (OBFL) flash.
BootupPortLoopback	Runs only during module bootup. Tests the packet path from the Supervisor CPU to each physical front panel port on the ASIC.
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 17: 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.

Diagnostic	Default Interval	Default Setting	Description
Module			
PortLoopback	On demand [for releases prior to Cisco NX-OS 7.0(3)I1(2)] 30 minutes [starting with Cisco NX-OS Release 7.0(3)I1(2)]	active	Checks diagnostics on a per-port basis on all admin down ports.
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.

Diagnostic	Default Interval	Default Setting	Description
Module			
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:

- The following Cisco Nexus switches and line cards do not support the run-time PortLoopback test but do support the BootupPortLoopback test:

  Switches

  - Cisco Nexus 92160YC-X

- Cisco Nexus 92304QC
- Cisco Nexus 9264PQ
- Cisco Nexus 9272Q
- Cisco Nexus 9232C
- Cisco Nexus 9236C
- Cisco Nexus 9256PV
- Cisco Nexus 92300YC
- Cisco Nexus 93108TC-EX
- Cisco Nexus 93108TC-EX-24
- Cisco Nexus 93180LC-EX
- Cisco Nexus 93180YC-EX
- Cisco Nexus 93180YC-EXU
- Cisco Nexus 93180YC-EX-24

Line Cards

- Cisco Nexus 9736C-EX
- Cisco Nexus 97160YC-EX
- Cisco Nexus 9732C-EX
- Cisco Nexus 9732C-EXM

- 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.
- The PortLoopback test is periodic, so the packet counter is incremented on admin down ports every 30 minutes. The test runs only on admin down ports. When a port is unshut, the counters are not affected.

Default Settings for Online Diagnostics

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

Parameters	**Default**
Bootup diagnostics level	complete
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** | **minimal** | **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. • **minimal**—Perform a minimal set of bootup diagnostics for the supervisor engine and bootup port loopback tests. • **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 14

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 contains 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:

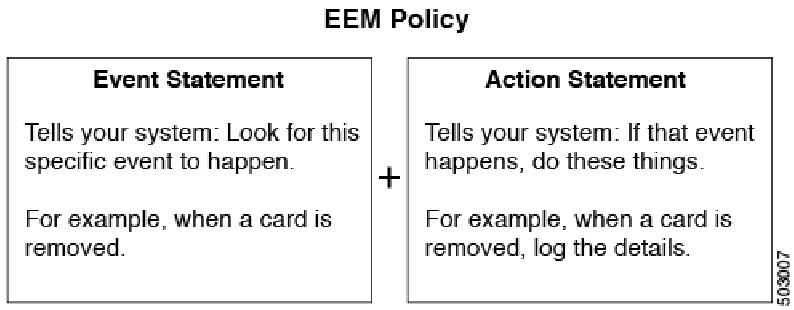
- Event statements—Events to monitor from another Cisco NX-OS component that may require some action, workaround, or notification.
- Action statements—An action that EEM can take, such as executing CLI commands, sending an email through the use of Smart Call Home feature, and disabling an interface to 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 3: 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.

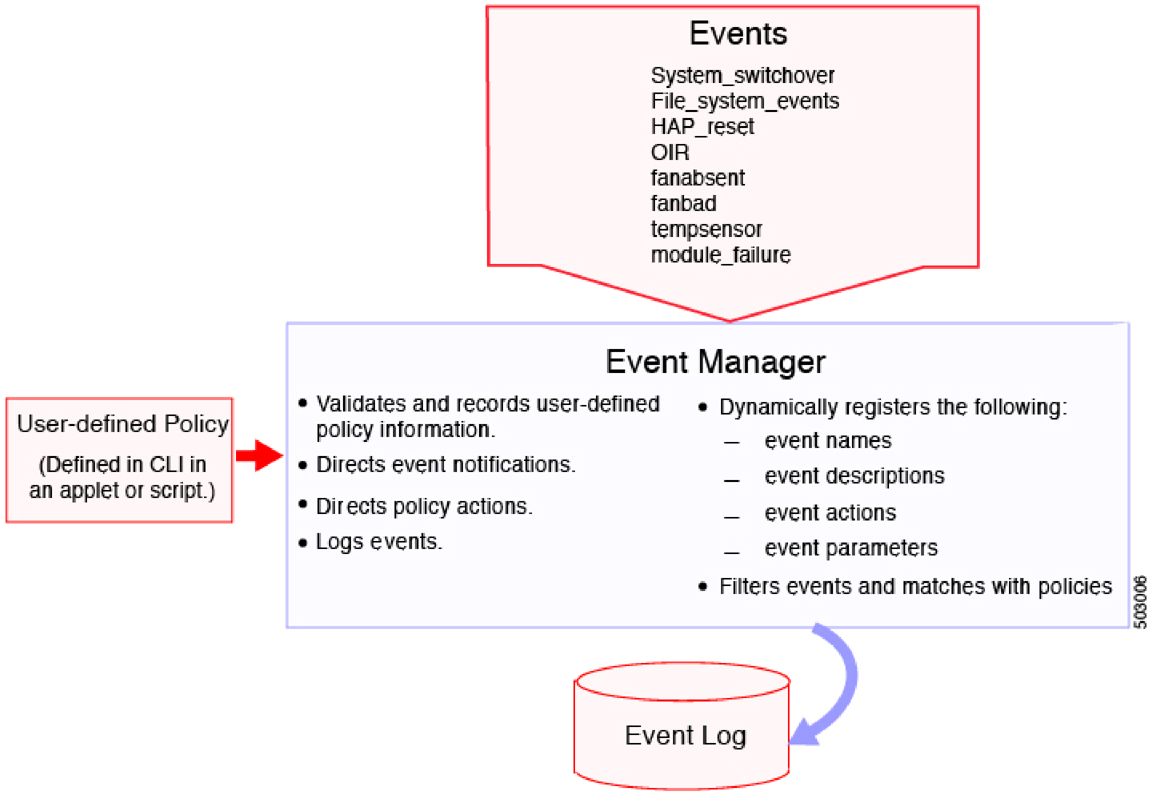
Event Statements

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 4: 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.
- 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.
- Note the following about override policies:
    - An override policy that consists of an event statement without an 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.

- Note the following about EEM event correlation:
  - 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.
  - EEM event correlation does not override the system default policies.

- 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.
- 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*	Creates an environment variable for EEM. The *variable-name* can be any case-sensitive,

	Command or Action	Purpose
	Example: `switch(config)# event manager environment emailto "admin@anyplace.com"`	alphanumeric string up to 29 characters. The *variable-value* can be any quoted alphanumeric string up to 39 characters.
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 204.
Step 5	(Optional) **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*}	Correlates multiple events in the policy.

	Command or Action	Purpose
	Example: `switch(config-applet)# tag one or two happens 1 in 10000`	The range for the *occurs* argument is from 1 to 4294967295. The range for the *seconds* argument is from 0 to 4294967295 seconds.
Step 6	**action** *number*[.*number2*] *action-statement* **Example:** `switch(config-applet)#action 1.0 cli show interface Ethernet 3/1`	Configures an action statement for the policy. Repeat this step for multiple action statements. See Configuring Action Statements, on page 209.
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 213.
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 Ethernet 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 Embedded Event Manager System Events and Configuration Examples, on page 385.

CHAPTER 15

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 that 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 that are 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 223.
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* **internal reset-reason**	Displays OBFL information for a specific module. **Note** If you specify **internal reset-reason** and you are operating in a redundant supervisor configuration, checking the persistent log on the standby supervisor after a system reset occurs will display a relevant reset reason. The reset reason is recorded on the on-board flash for both the active and standby supervisor.
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 16

Configuring SPAN

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

This chapter contains the following sections:

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)
- The inband interface to the control plane CPU

Note When you specify the supervisor inband interface as a SPAN source, the device monitors all packets that are sent by the Supervisor CPU.

- VLANs

- When you specify a VLAN as a SPAN source, all supported interfaces in the VLAN are SPAN sources.
- VLANs can be SPAN sources only in the ingress direction.

Note This applies to all switches except Cisco Nexus 9300-EX/-FX/-FX2 platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards.

- Satellite ports and host interface port channels on the Cisco Nexus 2000 Series Fabric Extender (FEX)
  - These interfaces are supported in Layer 2 access mode and Layer 2 trunk mode. They are not supported in Layer 3 mode, and Layer 3 subinterfaces are not supported.
  - 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.

Note A single SPAN session can include mixed sources in any combination of the above.

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.
- If you use the supervisor inband interface as a SPAN source, all packets generated by the supervisor hardware (egress) are monitored.

Note Rx is from the perspective of the ASIC (traffic egresses from the supervisor over the inband and is received by the ASIC/SPAN).

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
- Port channels in either access or trunk mode
- CPU as destination port
- Uplink ports on Cisco Nexus 9300 Series switches

Note FEX ports are not supported as SPAN destination ports.

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.
- The same destination interface cannot be used for multiple SPAN sessions. However, an interface can act as a destination for a SPAN and an ERSPAN session.
- Destination ports do not participate in any spanning tree instance. SPAN output includes bridge protocol data unit (BPDU) Spanning Tree Protocol hello packets.

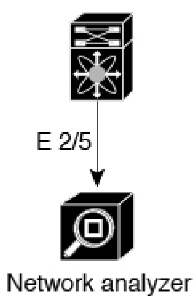
SPAN Sessions

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 5: 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.

Note A SPAN session with a VLAN source is not localized.

SPAN Truncation

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure the truncation of source packets for each SPAN session based on the size of the MTU. Truncation helps to decrease SPAN bandwidth by reducing the size of monitored packets. Any SPAN packet that is larger than the configured MTU size is truncated to the given size. For example, if you configure the MTU as 300 bytes, the packets with greater than 300 bytes are truncated to 300 bytes.

SPAN truncation is disabled by default. To use truncation, you must enable it for each SPAN session.

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:

- A maximum of 48 source interfaces are supported per SPAN session (Rx and Tx, Rx, or Tx).
- 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.
- 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.

- Packets with FCS errors are not mirrored in a SPAN session.
- You can configure only one destination port in a SPAN session.
- 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 can be replicated to the SPAN destination port although 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.
- 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.
- The combination of VLAN source session and port source session is not supported. If the traffic stream matches the VLAN source session and port source session, two copies are needed at two destination ports. Due to the hardware limitation, only the VLAN source SPAN and the specific destination port receive the SPAN packets. This limitation applies only to the following Cisco devices:

Table 18: Cisco Nexus 9000 Series Switches

Cisco Nexus 93120TX	Cisco Nexus 93128TX	Cisco Nexus 9332PQ

Cisco Nexus 9372PX	Cisco Nexus 9372PX-E	Cisco Nexus 9372TX
Cisco Nexus 9396PX	Cisco Nexus 9372TX-E	Cisco Nexus 9396TX

Table 19: Cisco Nexus 9000 Series Line Cards, Fabric Modules, and GEM Modules

N9K-X9408PC-CFP2	N9K-X9536PQ	N9K-C9508-FM
N9K-X9432PQ	N9K-X9564PX	N9K-C9504-FM
N9K-X9464PX	N9K-X9564TX	N9K-C9516-FM
N9K-X9464TX	N9K-X9636PQ	N9K-M4PC-CFP2

- 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.
- When you filter a monitor session, make sure that the access-group specified must be a VACL, or VLAN access-map and not a regular ACL for filtering purpose. This guidelines is not applicable for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- Supervisor-generated stream of bytes module header (SOBMH) packets have all 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.
- 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.
- SPAN copies for multicast packets are made before rewrite. Therefore, the TTL, VLAN ID, any remarking due to an 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 a Tx mirrored packet has a VLAN ID of 4095 on Cisco Nexus 9300 platform switches (except -EX, -FX, or -FX2) and Cisco Nexus 9500 platform modular switches.
- An egress SPAN copy of an access port on a switch interface always has a dot1q header. This guideline does not apply for Cisco Nexus 9508 platform 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 not to 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.
- VLAN sources are spanned only in the Rx direction. This limitation does not apply to the following switch platforms which support VLAN spanning in both directions:
    - Cisco Nexus 9300-EX platform switches
    - Cisco Nexus 9300-FX platform switches
    - Cisco Nexus 9300-FX2 platform switches

- Cisco Nexus 9504, 9508, and 9516 switches with the 97160YC-EX line card.
- Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.

- If a VLAN source is configured as both directions in one session and the physical interface source is configured in two other sessions, Rx SPAN is not supported for the physical interface source session. This limitation applies to the Cisco Nexus 97160YC-EX line card.
- 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 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.
- 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.
- Truncation is supported only for local and ERSPAN source sessions. It is not supported for ERSPAN destination sessions.
- Configuring MTU on a SPAN session truncates all 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.

SPAN Limitations for the Cisco Nexus 3000 Platform Switches

The following guidelines and limitations apply only the Nexus 3000 Series switches running Cisco Nexus 9000 code:

- The Cisco Nexus 3232C and 3264Q switches do not support SPAN on CPU as destination.

SPAN Limitations for the Cisco Nexus 9200 Platform Switches (excluding 9232E-B1)

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

The following guidelines and limitations apply only the Cisco Nexus 9200 platform switches:

- For Cisco Nexus 9200 platform switches, Rx SPAN is not supported for multicast without a forwarding interface on the same slice as the SPAN destination port.

- Tx SPAN for multicast, unknown multicast, and broadcast traffic are not supported on the Cisco Nexus 9200 platform switches.
- Tx SPAN of CPU-generated packets is not supported on Cisco Nexus 9200 platform switches.
- UDF-based SPAN is supported on the Cisco Nexus 9200 platform switches.
- The Cisco Nexus 9200 platform switches do not support Multiple ACL filters on the same source.
- VLAN Tx SPAN is supported on the Cisco Nexus 9200 platform switches.
- When multiple egress ports on the same slice are congested by egressing SPAN traffic, those egress ports will not get the line rate on the Cisco Nexus 9200 platform switches.
- Using the ACL filter to span subinterface traffic on the parent interface is not supported on the Cisco Nexus 9200 platform switches.
- On the Cisco Nexus 9200 platform switches, the CPU SPAN source can be added only for the Rx direction (SPAN packets coming from the CPU).
- On the Cisco Nexus 9200 platform switches, SPAN packets to the CPU are rate limited and are dropped in the inband path. You can change the rate limit using the **hardware rate-limiter span** command. You can analyze SPAN copies on the supervisor using the **ethanalyzer local interface inband mirror detail** command.

SPAN Limitations for the Cisco Nexus 9300 Platform Switches

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

The following guidelines and limitations apply only the Cisco Nexus 9300 platform switches:

- The following filtering limitations apply to egress (Tx) SPAN on all Cisco Nexus 9300-EX/FX/FX2 platform switches:
  - 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
- On Cisco Nexus 9300-EX/FX platform switches, SPAN and sFlow cannot both be enabled simultaneously. If one is active, the other cannot be enabled. However, on Cisco Nexus 9300-EX/FX/FX2 platform switches, both NetFlow and SPAN can be enabled simultaneously, providing a viable alternative to using sFlow and SPAN.

  Note Cisco Nexus 9300-FX2 switches support sFlow and SPAN co-existence.

- VLAN Tx SPAN is supported on Cisco Nexus 9300-EX and FX platform switches
- Cisco Nexus 9300 platform switches support multiple ACL filters on the same source.

- A single forwarding engine instance supports four SPAN sessions. For Cisco Nexus 9300 platform switches, if the first three sessions have bidirectional sources, the fourth session has hardware resources only for Rx sources.
- Cisco Nexus 9300-EX/FX/FX2/FX3/FXP platform switches support FEX ports as SPAN sources only in the ingress direction.
- Cisco Nexus 9300 platform switches (excluding Cisco Nexus 9300-EX/FX/FX2/FX3/FXP 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.
- Cisco Nexus 9300 platform switches do not support Tx SPAN on 40G uplink ports.

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

- Tx SPAN of CPU-generated packets is not supported on Cisco Nexus 9200, 9300-EX/FX/FXP/FX2/FX3/GX/GX2, 9300C, C9516-FM-E2, and C9508-FM-E2 switches.
- Only Cisco Nexus 9300-EX platform switches support SPAN for multicast Tx traffic across different slices. The slices must be on the same leaf spine engine (LSE).
- For Tx interface SPAN with Layer 2 switch port and port-channel sources on Cisco Nexus 9300-EX/FX/FX2 platform switches, only one copy is made per receiver unit regardless of how many Layer 2 members are receiving the stream in the same VLAN. For example, if e1/1-8 are all Tx direction SPAN sources and all are joined to the same group, the SPAN destination port sees one pre-rewrite copy of the stream, not eight copies. In addition, if for any reason one or more of those ports drops the packets on egress (for example, due to congestion), the packets may still reach the SPAN destination port. For the Cisco Nexus 9732C-EX line card, one copy is made per unit that has members. For port-channel sources, the Layer 2 member that will SPAN is the first port-channel member.
- SPAN Tx broadcast and SPAN Tx multicast are supported for Layer 2 port and port-channel sources across slices on Cisco Nexus 9300-EX/FX/FX2 platform switches, and the Cisco Nexus 9732C-EX line card, but only when IGMP snooping is disabled. (Otherwise, the slice limitation still applies.) These features are not supported for Layer 3 port sources, FEX ports (with unicast or multicast traffic), and VLAN sources.
- For SPAN Tx multicast for Layer 2, SPAN copies are created independent of multicast replication. Due to this, multicast and SPAN packet have different values for VLAN tag, which is the ingress interface VLAN ID.
- A SPAN copy of Cisco Nexus 9300 platform 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.

- UDF-based SPAN is supported on the Cisco Nexus 9300-EX/FX/FX2 platform switches.

- UDF-SPAN acl-filtering only supports source interface rx. This limitation applies to the following switches:
  - Cisco Nexus 9332PQ
  - Cisco Nexus 9372PX
  - Cisco Nexus 9372PX-E
  - Cisco Nexus 9372TX
  - Cisco Nexus 9372TX-E
  - Cisco Nexus 93120TX
- The Cisco Nexus 9300-EX/FX/FX2 platform switches do not support Multiple ACL filters on the same source.
- When multiple egress ports on the same slice are congested by egressing SPAN traffic, those egress ports will not get the line rate on the Cisco Nexus 9300-EX/FX/FX2 platform switches.
- Using the ACL filter to span subinterface traffic on the parent interface is not supported on the Cisco Nexus 9300-EX/FX/FX2 platform switches.
- On the Cisco Nexus 9300-EX/FX/FX2 platform switches, the CPU SPAN source can be added only for the Rx direction (SPAN packets coming from the CPU).
- On the Cisco Nexus 9300-EX/FX/FX2 platform switches, SPAN packets to the CPU are rate limited and are dropped in the inband path. You can change the rate limit using the **hardware rate-limiter span** command. You can analyze SPAN copies on the supervisor using the **ethanalyzer local interface inband mirror detail** command.

SPAN Limitations for the Cisco Nexus 9500 Platform Switches

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

The following guidelines and limitations apply only the Cisco Nexus 9500 platform switches:

- The following filtering limitations apply to egress (Tx) SPAN on 9500 platform switches with EX or FX line cards:
  - 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
- FEX and SPAN port-channel destinations are not supported on the Cisco Nexus 9500 platform switches with EX or FX line cards.
- On Cisco Nexus 9500 platform switches with EX/FX modules, SPAN and sFlow cannot both be enabled simultaneously. If one is active, the other cannot be enabled. However, on the Cisco Nexus 9500 platform

switches with EX or FX line cards, NetFlow and SPAN can both be enabled simultaneously, providing a viable alternative to using sFlow and SPAN.

- Cisco Nexus 9500 platform switches support VLAN Tx SPAN with the following line cards:
    - Cisco Nexus 97160YC-EX
    - Cisco Nexus 9732C-EX
    - Cisco Nexus 9732C-FX
    - Cisco Nexus 9736C-EX
    - Cisco Nexus 9736C-FX
    - Cisco Nexus 9736Q-FX
    - Cisco Nexus 9788TC-FX
- Cisco Nexus 9500 platform switches support multiple ACL filters on the same source.
- Tx SPAN of CPU-generated packets is not supported on Cisco Nexus 9500 platform switches with EX-based line cards.
- 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.

- On the Cisco Nexus 9500 platform switches, depending on the SPAN source's forwarding engine instance mappings, a single forwarding engine instance may support four SPAN sessions. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- Same source interface cannot be configured in multiple SPAN sessions on N9K-X96136YC-R line card.
- Multiple ACL filters are not supported on the same source.
- Cisco Nexus 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.
- SPAN does not support destinations on Cisco Nexus 9408PC-CFP2 line card ports.
- VLANs can be SPAN sources in the ingress and egress direction on Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- UDF-SPAN acl-filtering only supports source interface rx. This limitation applies to the following line cards:
    - Cisco Nexus 9564PX

- Cisco Nexus 9464TX2
- Cisco Nexus 9464TX
- Cisco Nexus 9464TX2
- Cisco Nexus 9564TX
- Cisco Nexus 9464PX
- Cisco Nexus 9536PQ
- Cisco Nexus 9636PQ
- Cisco Nexus 9432PQ

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.	—
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*[**rx** | **tx**] [**shut**]
	Example:
	`switch(config)# monitor session 3 rx`
	`switch(config-monitor)#`
	Example:
	`switch(config)# monitor session 3 tx`
	`switch(config-monitor)#`
	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.

	Command or Action	Purpose

Step 9	**source** {**interface** *type* [**rx** | **tx** | **both**] | [**vlan** {*number* | *range*}[**rx**]} | [**vsan** {*number* | *range*}[**rx**]}
	
	Example:
	`switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx`
	
	Example:
	`switch(config-monitor)# source interface port-channel 2`
	
	Example:
	`switch(config-monitor)# source interface sup-eth 0 rx`
	
	Example:
	`switch(config-monitor)# source vlan 3, 6-8 rx`
	
	Example:
	`switch(config-monitor)# source interface ethernet 101/1/1-3`
		Configures sources and the traffic direction in which to copy packets. You can enter a range of Ethernet ports, a port channel, an inband interface, a range of VLANs, or a satellite port or host interface port channel on the Cisco Nexus 2000 Series Fabric Extender (FEX).
		
		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.
		
		Note Source VLANs are supported only in the ingress direction. Source FEX ports are supported in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic.
		
		This note does not aply to Cisco Nexus 9300-EX/-FX/-FX2 series platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards.
		
		Supervisor as a source is only supported in the Rx direction.
		
		For a unidirectional session, the direction of the source must match the direction specified in the session.

Step 10	(Optional) Repeat Step 9 to configure all SPAN sources.	
Step 11	**filter vlan** {*number* | *range*} **Example:** `switch(config-monitor)# filter vlan 3-5, 7`	Configures which VLANs to select from the configured sources. You can configure one or more VLANs, as either a series of comma-separated entries or a range of numbers **Note** A FEX port that is configured as a SPAN source does not support VLAN filters.
Step 12	(Optional) Repeat Step 11 to configure all source VLANs to filter.	
Step 13	(Optional) **filter access-group** *acl-filter* **Example:** `switch(config-monitor)# filter access-group ACL1`	Associates an ACL with the SPAN session.

	Command or Action	Purpose

Step 14	Required: **destination interface** *type slot*/*port*
	Example:
	`switch(config-monitor)# destination interface ethernet 2/5`
	Example:
	`switch(config-monitor)# destination interface sup-eth 0`
		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.
		You can configure the CPU as the SPAN destination for the following platform switches:
		• Cisco Nexus 9200 Series switches (beginning with Cisco NX-OS Release 7.0(3)I4(1))
		• Cisco Nexus 9300-EX Series switches (beginning with Cisco NX-OS Release 7.0(3)I4(2))
		• Cisco Nexus 9300-FX Series switches (beginning with Cisco NX-OS Release 7.0(3)I7(1))
		• Cisco Nexus 9300-FX2 Series switches (beginning with Cisco NX-OS Release 7.0(3)I7(3))
		• Cisco Nexus 9500-EX Series switches with -EX/-FX line cards
		To do so, enter **sup-eth 0** for the interface type.

Step 15	Required: **no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 16	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-monitor)# show monitor session 3`	Displays the SPAN configuration.
Step 17	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring UDF-Based SPAN

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the SPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

Before you begin

Make sure that the appropriate TCAM region (racl, ifacl, or vacl) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based SPAN. For more information, see the "Configuring ACL TCAM Region Sizes" section in the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Procedure

	Command or Action	Purpose

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

Step 2	**udf** *udf-name offset-base offset length*
	
	Example:
	`switch(config)# udf udf-x packet-start 12 1`
	`switch(config)# udf udf-y header outer l3 20 2`
		Defines the UDF as follows:
		• *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
		• *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start | header** {**outer | inner** {**l3** | **l4**}}.
		• *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
		• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.
		
		You can define multiple UDFs, but Cisco recommends defining only required UDFs.

Step 3	**hardware access-list tcam region** {**racl | ifacl | vacl** } **qualify** *qualifier-name* **Example:**	Attaches the UDFs to one of the following TCAM regions: • racl—Applies to Layer 3 ports.

	Command or Action	Purpose
	`switch(config)# hardware access-list tcam region racl qualify ing-l3-span-filter`	• ifacl—Applies to Layer 2 ports • vacl—Applies to source VLANs. You can attach up to 8 UDFs to a TCAM region. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information,see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.

Step 6	**ip access-list** *span-acl*
	Example:
	`switch(config)# ip access-list span-acl-udf-only`
	`switch(config-acl)#`
		Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.

Step 7	Enter one of the following commands: • **permit udf** *udf-name value mask* • **permit ip** *source destination* **udf** *udf-name value mask* **Example:** `switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F`	Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

	Command or Action	Purpose
	Example: `switch(config-acl)# permit ip 10.0.0./24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F`	
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 SPAN Truncation

You can configure truncation for local and SPAN source sessions only.

Procedure

	Command or Action	Purpose

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

Step 2	**monitor session** *session number*
	Example:
	`switch(config)# monitor session 5`
	`switch(config-monitor)#`
		Enters monitor configuration mode for the specified SPAN session.

Step 3	**source interface** *type slot*/*port* [**rx** | **tx** | **both**] **Example:** `switch(config-monitor)# source interface ethernet 1/5 both`	Configures the source interface.

Step 4	**mtu** *size*
	Example:
	`switch(config-monitor)# mtu 320`
	Example:
	`switch(config-monitor)# mtu ?`
	`<320-1518> Enter the value of MTU truncation size for SPAN packets`
		Configures the MTU size for truncation. Any SPAN packet that is larger than the configured MTU size is truncated to the configured size. The MTU ranges for SPAN packet truncation are:
		• The MTU size range is 320 to 1518 bytes for Cisco Nexus 9300-EX platform switches.
		• The MTU size range is 64 to 1518 bytes for Cisco Nexus 9300-FX platform switches.

	Command or Action	Purpose
Step 5	**destination interface** *type slot*/*port* **Example:** `switch(config-monitor)# destination interface Ethernet 1/39`	Configures the Ethernet SPAN destination port.
Step 6	**no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 7	(Optional) **show monitor session** *session* **Example:** `switch(config-monitor)# show monitor session 5`	Displays the SPAN configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config-monitor)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SPAN for Multicast Tx Traffic Across Different LSE Slices

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure SPAN for multicast Tx traffic across different leaf spine engine (LSE) slices on Cisco Nexus 9300-EX platform switches.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **hardware multicast global-tx-span** **Example:** `switch(config)# hardware multicast global-tx-span`	Configures SPAN for multicast Tx traffic across different leaf spine engine (LSE) slices.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the device.

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.

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.

	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.

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)# source interface sup-eth 0 both
switch(config-monitor)# source vlan 3, 6-8 rx
switch(config-monitor)# source interface ethernet 101/1/1-3
switch(config-monitor)# filter vlan 3-5, 7
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 rx
switch(config-monitor)# filter vlan 3-5, 7
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)# vlan access-map span_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 span_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
switch(config-erspan-src)# filter access_group span_filter
```

Configuration Examples for UDF-Based SPAN

This example shows how to configure UDF-based SPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region racl qualify ing-l3-span-filter
copy running-config startup-config
reload
ip access-list acl-udf
permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1
source interface Ethernet 1/1
filter access-group acl-udf
```

This example shows how to configure UDF-based SPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)
- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l4 26 2
udf udf_pktsig_lsb header outer l4 28 2
hardware access-list tcam region racl qualify ing-l3-span-filter
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1
source interface Ethernet 1/1
filter access-group acl-udf-pktsig
```

Configuration Example for SPAN Truncation

This example shows how to configure SPAN truncation for use with MPLS stripping:

```
mpls strip
ip access-list mpls
statistics per-entry
20 permit ip any any redirect Ethernet1/5
interface Ethernet1/5
switchport
switchport mode trunk
mtu 9216
no shutdown
monitor session 1
source interface Ethernet1/5 tx
mtu 64
destination interface Ethernet1/6
  no shut
```

Configuration Examples for Multicast Tx SPAN Across LSE Slices

This example shows how to configure multicast Tx SPAN across LSE slices for Cisco Nexus 9300-EX platform switches. It also shows sample output before and after multicast Tx SPAN is configured.

Before Multicast Tx SPAN Is Configured

```
switch# show interface eth1/15-16, ethernet 1/27 counters

-----------------------------------------
Port            InOctets    InUcastPkts
-----------------------------------------
Eth1/15           580928              0
Eth1/16              239              0
Eth1/27                0              0

-----------------------------------------
Port         InMcastPkts    InBcastPkts
-----------------------------------------
Eth1/15             9077              0
Eth1/16                1              0
Eth1/27                0              0

-----------------------------------------
Port           OutOctets   OutUcastPkts
-----------------------------------------
Eth1/15              453              0
Eth1/16           581317              0
Eth1/27                0              0

-----------------------------------------
Port        OutMcastPkts   OutBcastPkts
-----------------------------------------
Eth1/15                4              0
Eth1/16             9080              0
Eth1/27                0              0
```

Configuring Multicast Tx SPAN

```
switch(config)# hardware multicast global-tx-span
Warning: Global Tx SPAN setting changed, please save config and reload
switch(config)# copy running-config start-up config
```

```
[########################################] 100%
Copy complete.
switch(config)# reload
This command will reboot the system. (y/n)?  [n] y
```

After Multicast Tx SPAN Is Configured

```
switch# show interface eth1/15-16, eth1/27 counters

-----------------------------------------
Port           InOctets    InUcastPkts
-----------------------------------------
Eth1/15          392576              0
Eth1/16               0              0
Eth1/27               0              0
-----------------------------------------
Port        InMcastPkts    InBcastPkts
-----------------------------------------
Eth1/15            6134              0
Eth1/16               0              0
Eth1/27               0              0

-----------------------------------------
Port          OutOctets   OutUcastPkts
-----------------------------------------
Eth1/15               0              0
Eth1/16          392644              0
Eth1/27          417112              0

-----------------------------------------
Port       OutMcastPkts   OutBcastPkts
-----------------------------------------
Eth1/15               0              0
Eth1/16            6135              0
Eth1/27            6134              0
```

Additional References

Related Documents

Related Topic	Document Title
ACL TCAM regions	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
FEX	*Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 9000 Series Switches*

CHAPTER 17

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.

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. Another method is that the destination can be the analyzer itself, which needs to understand the ERSPAN encapsulation format to parse the packet and access the inner (SPAN copy) frame.

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)
- Port channels
- The inband interface to the control plane CPU

> **Note** When you specify the supervisor inband interface as a SPAN source, the device monitors all packets that are sent by the Supervisor CPU.

> **Note** If you use the supervisor inband interface as a SPAN source, all packets generated by the supervisor hardware (egress) are monitored.
>
> Rx is from the perspective of the ASIC (traffic egresses from the supervisor over the inband and is received by the ASIC/SPAN).

- VLANs
  - When a VLAN is specified as an ERSPAN source, all supported interfaces in the VLAN are ERSPAN sources.
  - VLANs can be ERSPAN sources only in the ingress direction, except for Cisco Nexus 9300-EX/-FX/-FX2 series platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards.

> **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.

> **Note** An ERSPAN session with a VLAN source is not localized

ERSPAN Truncation

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure the truncation of source packets for each ERSPAN session based on the size of the MTU. Truncation helps to decrease ERSPAN bandwidth by reducing the size of monitored packets. Any ERSPAN packet that is larger than the configured MTU size is truncated to the given size. For ERSPAN, an additional ERSPAN header is added to the truncated packet from 54 to 166 bytes depending on the ERSPAN header type. For example, if you configure the MTU as 300 bytes, the packets are replicated with an ERSPAN header size from 354 to 466 bytes depending on the ERSPAN header type configuration.

ERSPAN truncation is disabled by default. To use truncation, you must enable it for each ERSPAN session.

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:

- A maximum of 48 source interfaces are supported per ERSPAN session (Rx and Tx, Rx, or Tx).
- ERSPAN destination handles jumbo frames for MTU differently based on the platform. For the following Cisco Nexus 9300 platform switches and Cisco Nexus 9500 platform switches with supporting line cards, ERSPAN destination drops the jumbo frames:
    - Cisco Nexus 9332PQ
    - Cisco Nexus 9372PX
    - Cisco Nexus 9372PX-E
    - Cisco Nexus 9372TX
    - Cisco Nexus 9372TX-E
    - Cisco Nexus 93120TX
    - Cisco Nexus 9500 platform switches with the following 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 platform switches and Cisco Nexus 9500 platform switches with supporting line cards, ERSPAN truncates the packets at port MTU, and issues a TX Output error:

    - 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
    - Cisco Nexus 9500 platform switches with the following 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.
- 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.
- Control plane packets that are 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.
- 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 must 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 before rewrite. Therefore, the TTL, VLAN ID, any remarking due to egress policy, and so on, are not captured in the ERSPAN copy.
- The timestamp granularity of ERSPAN Type III sessions is not configurable through the CLI. It is 100 picoseconds and driven through PTP.
- ERSPAN works on default and nondefault VRFs, but ERSPAN marker packets work only on the default VRF.
- The same source can be part of multiple sessions.
- 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:

- 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.
  - 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:
  - 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.

Default Settings

The following table lists the default settings for ERSPAN parameters.

Table 20: Default ERSPAN Parameters

Parameters	Default
ERSPAN sessions	Created in the shut state
ERSPAN marker packet interval	100 milliseconds
Timestamp granularity of ERSPAN Type III sessions	100 picoseconds

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** or **Example:** `switch(config)# monitor erspan origin ip-address 10.0.0.1 global`	Configures the ERSPAN global origin IP address.

	Command or Action	Purpose
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.

Step 6	**source** {**interface** *type* [**tx** | **rx** |**both**] **vlan** {**number** | **range**} [**rx**]}
	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`
	Example:
	`switch(config-erspan-src)# source interface`
	`sup-eth 0 rx`
	Example:
	`switch(config-erspan-src)# source vlan 3, 6-8 rx`
	Example:
	`switch(config-erspan-src)# source interface`
	`ethernet 101/1/1-3`
		Configures the sources and traffic direction in which to copy packets. You can enter a range of Ethernet ports, a port channel, an inband interface, a range of VLANs, or a satellite port or host interface port channel on the Cisco Nexus 2000 Series Fabric Extender (FEX).
		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.
		Note Source VLANs are supported only in the ingress direction. Source FEX ports are supported in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic.
		Supervisor as a source is only supported in the Rx direction.

Step 7	(Optional) Repeat Step 7 to configure all ERSPAN sources.	—
Step 8	**filter vlan** {*number* | *range*} **Example:** `switch(config-erspan-src)# filter vlan 3-5, 7`	Configures which VLANs to select from the configured sources. You can configure one or more VLANs, as either a series of comma-separated entries or a range of numbers. For information on the VLAN range,

	Command or Action	Purpose
		see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide. **Note** A FEX port that is configured as an ERSPAN source does not support VLAN filters.
Step 9	(Optional) Repeat Step 9 to configure all source VLANs — to filter.	—
Step 10	(Optional) **filter access-group** *acl-filter* **Example:** `switch(config-erspan-src)# filter access-group ACL1`	Associates an ACL with the ERSPAN session. (You can create an ACL using the standard ACL configuration process. For more information, see the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*) **Note** Before executing this command, configure ip access list and associated vlan access mac. See Configuring an ERSPAN ACL.
Step 11	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1`	Configures the destination IP address in the ERSPAN session. **Note** Only one destination IP address is supported per ERSPAN source session.
Step 12	**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 13	**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 14	(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 15	(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 16	(Optional) [**no**] **marker-packet** *milliseconds* **Example:**	Enables the ERSPAN marker packet for a session in order to recover the real value of the ERSPAN timestamp. The interval can range

	Command or Action	Purpose
	`switch(config-erspan-src)# marker-packet 100`	from 100 to 1000 milliseconds. The **no** form of this command disables the marker packet for the session. **Note** ERSPAN marker packets only apply to Type 3 sessions.
Step 17	**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 18	**exit**
	Example:
	`switch(config-erspan-src)# exit`
	`switch(config)#`
		Exits the monitor configuration mode.

Step 19	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config)# show monitor session 3`	Displays the ERSPAN session configuration.
Step 20	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the running ERSPAN configuration.
Step 21	(Optional) **show startup-config monitor** **Example:** `switch(config)# show startup-config monitor`	Displays the ERSPAN startup configuration.
Step 22	(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 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:**	Displays the ERSPAN running configuration.

	Command or Action	Purpose
	`switch(config)# show running-config monitor`	
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.

Configuring an ERSPAN ACL

You can create an IPv4 ERSPAN ACL on the device and add rules to it.

Before you begin

To modify the DSCP value or the GRE protocol, you need to allocate a new destination monitor session. A maximum of four destination monitor sessions are supported.

Procedure

	Command or Action	Purpose

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

Step 2	**ip access-list** *acl-name*
	Example:
	`switch(config)# ip access-list erspan-acl`
	`switch(config-acl)#`
		Creates the ERSPAN ACL and enters IP ACL configuration mode. The *acl-name* argument can be up to 64 characters.

Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination* [**set-erspan-dscp** *dscp-value*] [**set-erspan-gre-proto** *protocol-value*] **Example:** `switch(config-acl)# permit ip 192.168.2.0/24 any set-erspan-dscp 40 set-erspan-gre-proto 5555` **Example:**	Creates a rule in the ERSPAN ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. The **set-erspan-dscp** option sets the DSCP value in the ERSPAN outer IP header. The range for the DSCP value is from 0 to 63. The DSCPvalue configured in the ERSPAN ACL

	Command or Action	Purpose

	`switch(config)# ip access-list match_11_pkts`
	`switch(config-acl)# permit ip 10.0.0.0/24 any`
	`switch(config-acl)# exit`
		overrides the value configured in the monitorsession. If you do not include this option in the ERSPAN ACL, 0 or the DSCP value configured in the monitor session will be set.
		
		The **set-erspan-gre-proto** option sets the protocol value in the ERSPAN GRE header. The range for the protocol value is from 0 to 65535. If you do not include this option in the ERSPAN ACL, the default value of 0x88be will be set as the protocol in the GRE header for ERSPAN-encapsulated packets
		
		Each access control entry (ACE) with the s**set-erspan-gre-proto** or **set-erspan-dscp** action consumes one destination monitor session. A maximum of three ACEs with one of these actions is supported per ERSPAN ACL. For example, you can configure one of the following:
		
		• One ERSPAN session with an ACL having a maximum of three ACEs with the **set-erspan-gre-proto** or **set-erspan-dscp** action
		
		• One ERSPAN session with an ACL having two ACEs with the **set-erspan-gre-proto** or **set-erspan-dscp** action and one additional local or ERSPAN session
		
		• A maximum of two ERSPAN sessions with an ACL having one ACE with the **set-erspan-gre-proto** or **set-erspan-dscp** action

Step 4	**show ip access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists erpsan-acl`	Displays the ERSPAN ACL configuration.
Step 5	**show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-acl)# show monitor session 1`	Displays the ERSPAN session configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring UDF-Based ERSPAN

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the ERSPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

Before you begin

Make sure that the appropriate TCAM region (racl, ifacl, or vacl) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based ERSPAN. For information, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Procedure

	Command or Action	Purpose

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

Step 2	**udf** *udf-name offset-base offset length*
	
	Example:
	`switch(config)# udf udf-x packet-start 12 1`
	`switch(config)# udf udf-y header outer l3 20 2`
		Defines the UDF as follows:
		• *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
		• *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}.
		• *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
		• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.
		
		You can define multiple UDFs, but Cisco recommends defining only required UDFs.

Step 3	**hardware access-list tcam region** {**racl** | **ifacl** | **vacl** }**qualify udf** *udf-names* **Example:**	Attaches the UDFs to one of the following TCAM regions: • racl—Applies to Layer 3 ports.—Applies to layer 2 and Layer 3 ports.

	Command or Action	Purpose
	`switch(config)# hardware access-list tcam region racl qualify udf udf-x udf-y`	• ifacl—Applies to Layer 2 ports. • vacl—Applies to source VLANs. You can attach up to 8 UDFs to a TCAM region. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.

Step 6	**ip access-list** *erspan-acl*
	
	Example:
	`switch(config)# ip access-list erspan-acl-udf-only`
	`switch(config-acl)#`
		Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.

Step 7	Enter one of the following commands: • **permit udf** *udf-name value mask* • **permit ip** *source destination* **udf** *udf-name value mask* **Example:** `switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F`	Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

	Command or Action	Purpose
	Example: `switch(config-acl)# permit ip 10.0.0.0/24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F`	
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 ERSPAN Truncation

You can configure truncation for local and ERSPAN source sessions only.

Procedure

	Command or Action	Purpose

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

Step 2	**monitor session** *session-number* **type erspan-source**
	Example:
	`switch(config)# monitor session 10 type erspan-source`
	`switch(config-erspan-src)#`
		Enters monitor configuration mode for the specified ERSPAN session.

Step 3	**source interface** *type slot/port* [**rx** | **tx** | **both**] **Example:** `switch(config-erspan-src)# source interface ethernet 1/5 both`	Configures the source interface.

Step 4	**mtu** *size*
	Example:
	`switch(config-erspan-src)# mtu 512`
	Example:
	`switch(config-erspan-src)# mtu ?`
	`<512-1518> Enter the value of MTU truncation size for ERSPAN packets (erspan header + truncated original packet)`
		Configures the MTU size for truncation. Any ERSPAN packet that is larger than the configured MTU size is truncated to the configured size. The MTU ranges for ERSPAN packet truncation are:
		• The MTU size range is 512 to 1518 bytes for Cisco Nexus 9300-EX platform switches.
		• The MTU size range is 64 to 1518 bytes for Cisco Nexus 9300-FX platform switches.

	Command or Action	Purpose
Step 5	**destination interface** *type slot*/*port* **Example:** `switch(config-erspan-src)# destination interface Ethernet 1/39`	Configures the Ethernet ERSPAN destination port.
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	(Optional) **show monitor session** *session* **Example:** `switch(config-erspan-src)# show monitor session 5`	Displays the ERSPAN configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config-erspan-src)# 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 ip access-lists *name*	Displays the ERSPAN ACL configuration.

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the ERSPAN session configuration. The output includes the egress interface that is used to send the ERSPAN packets. The output varies depending on the type of egress interface used: • Physical Layer 3 interface—Displays the interface name. • SVI interface—Displays the member interface through which the route was learned. • Layer 3 port channel—Displays the port-channel interface name. • Layer 3 subinterface—Displays the parent interface name. • ECMP path—Displays the name of one of the equal-cost multipath (ECMP) member interfaces. Only the interface that is displayed will be used for mirroring the traffic even though the route is ECMP. • PFC on interfaces—Displays the priority flow control (PFC) status on the interface.
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 a Unidirectional ERSPAN Session

This example shows how to configure a unidirectional ERSPAN session:

```
switch# configure terminal
switch(config)# interface ethernet 14/30
switch(config-if)# no shut
switch(config-if)# exit
switch(config)# no monitor session 3
switch(config)# monitor session 3 rxswitch(config-erspan-src)# source interface ethernet
2/1-3 rx
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# no shut
switch(config-erspan-src)# exit
switch(config)# show monitor session 1
```

Configuration Example for an ERSPAN ACL

This example shows how to configure an ERSPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_10_pkts
switch(config-acl)# permit ip 10.0.0.0/24 any
switch(config-acl)# exit
switch(config)# ip access-list match_172_pkts
switch(config-acl)# permit ip 172.16.0.0/24 any
switch(config-acl)# exit
```

In the case of different ERSPAN destinations where the interesting traffic is chosen based on the defined ACL filters, the last configured session would always have the higher priority.

For example, if Monitor Session 1 is configured; then Monitor Session 2 is configured; then ERSPAN traffic filter works as intended. But, if the user goes back to Monitor Session 1 and re-applies one of the existing configuration line (no new changes in the config); then the spanned traffic switches back to Monitor Session 1.

Configuration Example for a Marker Packet

This example shows how to enable the ERSPAN marker packet with an interval of 2 seconds:

```
switch# configure terminal
switch(config)# monitor erspan origin ip-address 172.28.15.250 global
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# header-type 3
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# source interface ethernet 1/15 both
switch(config-erspan-src)# marker-packet 100
switch(config-erspan-src)# no shut
switch(config-erspan-src)# show monitor session 1
session 1
---------------
type              : erspan-source
state             : up
granularity       : nanoseconds
erspan-id         : 1
vrf-name          : default
destination-ip    : 10.1.1.2
ip-ttl            : 16
ip-dscp           : 5
header-type       : 3
origin-ip         : 172.28.15.250 (global)
source intf       :
    rx            : Eth1/15
    tx            : Eth1/15
    both          : Eth1/15
    rx            :
marker-packet     : enabled
packet interval   : 100
packet sent       : 25
packet failed     : 0
egress-intf       :
```

Configuration Examples for UDF-Based ERSPAN

This example shows how to configure UDF-based ERSPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region racl qualify udf udf_tcpflags
copy running-config startup-config
reload
ip access-list acl-udf
permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1 type erspan-source
source interface Ethernet 1/1
filter access-group acl-udf
```

This example shows how to configure UDF-based ERSPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)
- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l3 26 2
udf udf_pktsig_lsb header outer l3 28 2
hardware access-list tcam region racl qualify udf udf_pktsig_msb udf_pktsig_lsb
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1 type erspan-source
source interface Ethernet 1/1
filter access-group acl-udf-pktsig
```

Configuration Example for ERSPAN Truncation

This example shows how to configure ERSPAN truncation for use with MPLS stripping:

```
mpls strip
ip access-list mpls
  statistics per-entry
```

```
  20 permit ip any any redirect Ethernet1/5

interface Ethernet1/5
  switchport
  switchport mode trunk
  mtu 9216
  no shutdown

monitor session 1
  source interface Ethernet1/5 tx
  mtu 64
  destination interface Ethernet1/6
  no shut
monitor session 21 type erspan-source
  description "ERSPAN Session 21"
  header-type 3
  erspan-id 21
  vrf default
  destination ip 10.1.1.2
  source interface Ethernet1/5 tx
  mtu 64
  no shut
monitor session 22 type erspan-source
  description "ERSPAN Session 22"
  erspan-id 22
  vrf default
  destination ip 10.2.1.2
  source interface Ethernet1/5 tx
  mtu 750
  no shut
monitor session 23 type erspan-source
  description "ERSPAN Session 23"
  header-type 3
  marker-packet 1000
  erspan-id 23
  vrf default
  destination ip 10.3.1.2
  source interface Ethernet1/5 tx
  mtu 1000
  no shut
```

CHAPTER 18

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 contains 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

About DCBXP

The Data Center Bridging Exchange Protocol (DCBXP) is an extension of LLDP. It is used to announce, exchange, and negotiate node parameters between peers. DCBXP parameters are packaged as DCBXP TLVs in the LLDP packet. If CEE is used, DCBXP will use an acknowledgment mechanism over LLDP. When the port comes up, DCBX TLVs are sent and any DCBX TLVs received are processed. By default, the DCBX protocol is set to auto-detect, and the latest protocol version supported by both the peers is used.

Features that need to exchange and negotiate parameters with peer nodes using DCBXP are as follows:

- Priority-based Flow Control (PFC)—PFC is an enhancement to the existing Pause mechanism in Ethernet. It enables Pause based on user priorities or classes of service. A physical link that is divided into eight virtual links with PFC provides the capability to use Pause on a single virtual link without affecting traffic on the other virtual links. Enabling Pause on a per-user-priority basis allows administrators to create lossless links for traffic requiring no-drop service while retaining packet-drop congestion management for IP traffic.
- Enhanced Transmission Selection (ETS)—ETS enables optimal bandwidth management of virtual links. ETS is also called priority grouping. It enables differentiated treatments within the same priority classes of PFC. ETS provides prioritized processing based on bandwidth allocation, low latency, or best effort, resulting in per-group traffic class allocation. For example, an Ethernet class of traffic may have a high-priority designation and a best effort within that same class. ETS allows differentiation between traffic of the same priority class, thus creating priority groups.
- Application Priority Configuration—Carries information about the priorities that are assigned to specific protocols.
- Priority to DSCP Mapping—The mapping of the DSCP and COS values configured in the QoS policy are sent in the Application Priority TLV.

Note For information on the quality of service (QoS) features, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

DCBXP is enabled by default, provided LLDP is enabled. When LLDP is enabled, DCBXP can be enabled or disabled using the [**no**] **lldp tlv-select dcbxp** command. DCBXP is disabled on ports where LLDP transmit or receive is disabled.

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 supported on the following platforms:
  - Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 Series switches
  - Cisco Nexus 9332C, 9332PQ, 9364C, 9372PX, 9372PX-E, and 9396PX switches
  - Cisco Nexus 9504 and 9508 switches with X9432PQ, X9464PX, X9536PQ, X9564PX, X9636PQ, X9732C-EX, and X9736C-FX, line cards
- The Cisco Nexus 3232C and 3264Q switches do not support DCBXP.
- DCBXP incompatibility messages might appear when you change the network QoS policy if a physical loopback connection is in the device. The incompatibility exists for only a short time and then clears.
- DCBX TLVs are sent only when the input queuing is configured and applied at the system level.
- PFC TLV are sent when pause is enabled for at-least one COS value in network-qos policy and priority-flow-control mode should be auto in the Interface level.

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 the DCBXP Protocol Version

You can specify the protocol version in which the DCBX TLVs are sent.

Note If the peers are not running the same version, DCBX parameters may not converge for the link. You may need to reset the link for the new protocol version to take effect.

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 1/25`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**lldp dcbx version** *cee*/*ieee*/*auto* **Example:** `switch(config-if)#lldp dcbx version cee`	Specifies the protocol version mode sent. • The *cee* variable sets the port to only send TLVs in Converged Enhanced Ethernet (CEE) protocol version. • The *ieee* variable sets the port to only sent TLVs in IEEE 802.1Qaz protocol version. • The *auto* variable sets the port to send TLVs in the latest protocol version supported by both the peers. The default is set to *auto*.

Multiple LLDP Neighbors Per Physical Interface

Often times a network device sends multiple LLDP packets, out of which one is from the actual host. If a Cisco Nexus switch is communicating with the device but can only manage a single LLDP neighbor per interface, there is a good chance that becoming a neighbor with the actual required host will fail. To minimize this, Cisco Nexus switch interfaces can support multiple LLDP neighbors creating a better opportunity of becoming an LLDP neighbor with the correct device.

Support for multiple LLDP neighbors over the same interface requires LLDP multi-neighbor support to be configured globally.

Note You must disable DCBX globally before configuring LLDP multi-neighbor support. Failure to do so invokes an error message.

Enabling or Disabling LLDP Multi-Neighbor Support

Before you begin

Consider the following before enabling LLDP multi-neighbor support on the interfaces:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

> **Note** After you globally enable LLDP, it is enabled on all supported interfaces by default.

- A maximum of three (3) neighbors are supported on an interface.
- LLDP multi-neighbor is not supported on FEX interfaces.

Procedure

	Command or Action	Purpose

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

Step 2	Required: **no lldp tlv-select dcbxp**
	Example:
	`switch(config)# no lldp tlv-select dcbxp`
	`switch(config)#`
		Disables DCBXP TLVs globally.
		Note This command must be entered to avoid invoking an error message once LLDP multi-neighbor support is configured.

Step 3	Required: [**no**] **lldp multi-neighbor**
	Example:
	`switch(config)# lldp multi-neighbor`
	`switch(config)#`
		Enables or disables LLDP multi-neighbor support for all interfaces globally.

Step 4	**interface** *port* / *slot*
	Example:
	`switch(config)# interface 1/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling LLDP and enters the interface configuration mode.

Step 5	(Optional) [**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Disables (or enables) the transmission of LLDP packets on the interface. **Note** The transmission of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.

	Command or Action	Purpose
Step 6	(Optional) [**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Disables (or enables) the reception of LLDP packets on the interface. **Note** The reception of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.
Step 7	(Optional) **show lldp interface** *port* / *slot* **Example:** `switch(config-if)# show lldp interface 1/1`	Displays the LLDP configuration on the interface.
Step 8	(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 Support on Port-Channel Interfaces

Before you begin

Consider the following before enabling LLDP support on port-channels:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

> **Note** After you globally enable LLDP, it is enabled on all supported interfaces by default.

- Applying the **lldp transmit** and **lldp receive** configuration commands to a port-channel does not affect the configuration for the members of the port-channel.
- LLDP neighbors form between the port-channels only when LLDP transmit and receive is configured on both sides of the port-channel.
- The LLDP transmit and receive commands do not work on MCT, VPC, fex-fabric, FEX port-channels, and port-channel sub-interfaces.

Note If you enable the LLDP port-channel feature globally, the LLDP configuration is not applied to any of these port types. If the configuration is removed from the port-channels or the port type feature is disabled globally, you cannot use the **lldp port-channel** command to enable it on the newly supported port-channels. The command was already issued. To enable LLDP port-channel on the port-channels in question, configure **lldp transmit** and **lldp receive** for each port-channel (see steps 4, 5, and 6 in the following procedure).

Procedure

	Command or Action	Purpose

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

Step 2	Required: **no lldp tlv-select dcbxp**
	Example:
	`switch(config)# no lldp tlv-select dcbxp`
	`switch(config)#`
		Disables DCBXP TLVs globally. You must enter this command before configuring LLDP on port-channels.

Step 3	Required: [**no**] **lldp port-channel**
	Example:
	`switch(config)# lldp port-channel`
	`switch(config)#`
		Enables or disables LLDP transmit and receive for all port channels globally.

Step 4	**interface port-channel** [*port-channel-number* | *port-channel-range*]
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
	Example:
	Enter a range of port-channel numbers if you are configuring LLDP over more than one port-channel:
	`switch(config)# interface port-channel 1-3`
	`switch(config-if-range)#`
		Specifies the interface port-channel on which you are enabling LLDP and enters the interface configuration mode.
		Specifies the interface port-channel range on which you are enabling LLDP and enters the interface range configuration mode.

Step 5	(Optional) [**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Disables (or enables) the transmission of LLDP packets on the port-channel or range of port-channels.

	Command or Action	Purpose
		Note The transmission of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.
Step 6	(Optional) [**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Disables (or enables) the reception of LLDP packets on the port-channel or range of port-channels. **Note** The reception of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.
Step 7	(Optional) **show lldp interface port-channel**l *port-channel-number* **Example:** `switch(config-if)# show lldp interface port-channel 3`	Displays the LLDP configuration on the port-channel.
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 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.

	Command or Action	Purpose
		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 dcbxp, 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 all	Displays the LLDP DCBXP, transmit and receive configuration for all interfaces.
show lldp interface *interface slot*/*port*	Displays the LLDP interface configuration.

Command	Purpose
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 dcbx interface *interface slot*/*port*	Displays DCBXP TLV information for a specific interface.
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.
show qos dcbxp *interface slot*/*port*	Displays DCBXP information for a specific 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 19

Configuring NetFlow

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

This chapter contains the following sections:

About NetFlow

NetFlow identifies packet flows for ingress IP packets and provides statistics based on these packet flows. NetFlow does not require any change to either the packets themselves or to any networking device.

NetFlow uses flows to provide statistics for accounting, network monitoring, and network planning. A flow is a unidirectional stream of packets that arrives on a source interface (or VLAN) and has the same values for the keys. A key is an identified value for a field within the packet. You create a flow using a flow record to define the unique keys for your flow.

Cisco NX-OS supports the flexible NetFlow feature that enables enhanced network anomalies and security detection. Flexible NetFlow allows you to define an optimal flow record for a particular application by selecting the keys from a large collection of predefined fields.

All key values must match for the packet to count in a given flow. A flow might gather other fields of interest, depending on the export record version that you configure. Flows are stored in the NetFlow cache.

You can export the data that NetFlow gathers for your flow by using a flow exporter and export this data to a remote NetFlow Collector, such as Cisco Stealthwatch. Cisco NX-OS exports a flow as part of a NetFlow export User Datagram Protocol (UDP) datagram under the following circumstances:

- Flows are exported periodically as per the flow timeout value, which defaults to 10 seconds if not configured.
- You have forced the flow to export.

The flow record determines the size of the data to be collected for a flow. The flow monitor combines the flow record and flow exporter with the NetFlow cache information.

Cisco NX-OS can gather NetFlow statistics and analyze all packets on the interface or subinterface.

Dual-Layer NetFlow Implementation

Unlike other Cisco Nexus platforms, Cisco Nexus 9000 Series switches separate NetFlow processing into two layers:

- The first layer supports per-packet visibility for line-rate traffic. Packets do not need to be sampled and statistically analyzed. Instead, the packets can be processed and aggregated at line rate.
- The second layer enables the gathering of flows at scale. It can maintain hundreds of thousands of flows without losing any flows and periodically exports them to an external collector.

Flow Records

A flow record defines the keys that NetFlow uses to identify packets and other fields of interest that NetFlow gathers for the flow. You can define a flow record with any combination of keys and fields of interest. Cisco NX-OS supports a rich set of keys. A flow record also defines the types of counters gathered per flow. You can configure 32- or 64-bit packet or byte counters.

The key fields are specified with the **match** keyword. The fields of interest and counters are specified under the **collect** keyword.

Cisco NX-OS enables the following match fields as the defaults when you create a flow record:

- match interface input
- match flow direction

Flow Exporters

A flow exporter contains network layer and transport layer details for the NetFlow export packet. You can configure the following information in a flow exporter:

- Export destination IP address
- Source interface
- UDP port number (where the NetFlow Collector is listening for NetFlow packets)—The default value is 9995.

Note NetFlow export packets use the IP address that is assigned to the source interface. If the source interface does not have an IP address assigned to it, the flow exporter drops flows that were meant to be exported.

Cisco NX-OS exports data to the NetFlow Collector whenever a timeout occurs. You can configure a flush cache timeout (using the **flow timeout** command) to flush the cache and force a flow export.

Export Format

Cisco NX-OS supports the Version 9 export format. This format supports a more efficient network utilization than the older Version 5 export format and supports IPv6 and Layer 2 fields. In addition, the Version 9 export format supports the full 32-bit SNMP ifIndex values at the NetFlow Collector.

Layer 2 NetFlow Keys

You can define Layer 2 keys in flexible NetFlow records that you can use to capture flows in Layer 2 interfaces. The Layer 2 keys are as follows:

- Source and destination MAC addresses
- Source VLAN ID
- EtherType from the Ethernet frame

You can apply Layer 2 NetFlow to the following interfaces for the ingress direction:

- Switch ports in access mode
- Switch ports in trunk mode
- Layer 2 port channels

Note You cannot apply Layer 2 NetFlow to VLANs, egress interfaces, or Layer 3 interfaces such as VLAN interfaces.

Flow Monitors

A flow monitor references the flow record and flow exporter. You apply a flow monitor to an interface.

High Availability

Cisco NX-OS supports stateful restarts for NetFlow. After a reboot, Cisco NX-OS applies the running configuration.

The flow cache is not preserved across restarts, and packets that come to the software during restarts cannot be processed.

Prerequisites for NetFlow

NetFlow has the following prerequisites:

- Make sure that you understand the resources required on your device because NetFlow consumes memory and CPU resources.

Guidelines and Limitations for NetFlow

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

NetFlow has the following configuration guidelines and limitations:

- For Cisco Nexus 9300-FX platform switches only, if you add a member to a port channel that is already configured for Layer 2 NetFlow, its NetFlow configuration is removed and the Layer 2 configuration of the port channel is added to it.
- NetFlow is not supported on tunnel interfaces.
- NetFlow is not supported for CPU-transmitted packets.
- Only ingress NetFlow is supported. Egress NetFlow is not supported.
- Flow cache can be cleared per flow type, such as Layer 2, IPv4, and IPv6. It cannot be cleared per flow monitor.
- Flow collection is not performed for ARP traffic.
- You must configure a source interface for the NetFlow Data Export (NDE). If you do not configure a source interface, the flow exporter drops flows that were meant to be exported.
- Layer 2 switched flow monitors are applied only to Layer 2 interfaces. IP and IPv6 flow monitors can be applied to VLANs, SVIs, Layer 3 routed interfaces, or subinterfaces.
- If you change a Layer 2 interface to a Layer 3 interface, or a Layer 3 interface to a Layer 2 interface, the software removes the Layer 2 NetFlow configuration from the interface.
- The same flow monitor cannot be shared with a VLAN and Layer 3 interfaces (for example, physical Layer 3 interface, SVI interface, or Layer 3 subinterface). You must distinguish a VLAN and Layer 3 interface since the ACL is different and cannot be shared. They must be treated as two different profiles.
- A rollback fails if you try to modify a record that is programmed in the hardware during a rollback.
- The limitations of the NetFlow feature are as follows:
  - NetFlow for MPLS/VXLAN datapath is not supported
  - NetFlow is not supported on loopback and switch management interfaces.
- The following guidelines and limitations are applicable to Netflow in a VXLAN environment:
  - NetFlow is supported on SVI and non-uplink L3 Interfaces of a VXLAN VTEP. This does not include the L3VNI SVI.
  - NetFlow is not supported on uplink interfaces on a VXLAN VTEP.
  - NetFlow on Multisite Border Gateways is not supported.
  - A NetFlow Collector that is reachable over the VXLAN fabric is supported.
- Beginning with Cisco NX-OS Release 9.2(1):

  - NetFlow for FEX Layer 3 ports is supported on Cisco Nexus 9300-EX and 9300-FX platform switches.
  - NetFlow CE is supported on the Cisco Nexus 9300-EX platform switches.

Note All EX type platform switches, including the Cisco Nexus 9700-EX line cards, CE NetFlow only captures CE flow records for non-IPv4 and IPv6 traffic flows. Whereas for FX and FX2 type platform switches and line cards, we can capture CE flow data for IP flows as long as **mac packet-classify** is applied on the interface.

- Beginning with Cisco NX-OS Release 9.2(2), the Cisco Nexus 9300-FX switch supports collecting the OUTPUT_SNMP field for NetFlow Data Export (NDE). No other Cisco Nexus 9000 platform switch or Cisco Nexus line card supports collecting the OUTPUT_SNMP field.
- Beginning with Cisco NX-OS Release 9.2(2), NetFlow is supported on Cisco Nexus 9500 platform switches with Cisco Nexus 9700-EX line cards and FM-E modules.
- NetFlow is not supported on Cisco Nexus 92348GC-X platform switch.
- For Cisco Nexus 9300-EX platform switches, a flow monitor applied on a VLAN or SVI can collect flows for both switched and routed traffic. For Cisco Nexus 9300-FX platform switches, NetFlow VLANs are supported for switched traffic only, and NetFlow SVIs are supported for routed traffic only.
- The Cisco Nexus 9300-EX platform switch supports NetFlow and SPAN on the same interface at the same time. This functionality is a viable alternative to using SPAN and sFlow.
- On Cisco Nexus 9300-EX/FX platform switches, and Cisco Nexus 9500 platform switches with EX/FX modules, SPAN, and sFlow cannot both be enabled simultaneously. If one is active, the other cannot be enabled. However, on the Cisco Nexus 9300-EX/FX/FX2 and the Cisco Nexus 9500 platform switches with EX modules, both NetFlow and SPAN can be enabled simultaneously, providing a viable alternative to using sFlow and SPAN.

Note Cisco Nexus 9300-FX2 platform switches support sFlow and SPAN coexistence.

- For Cisco Nexus 9300-EX platform switches, the same flow monitor cannot be attached to a VLAN and an SVI at the same time.
- The Cisco Nexus 9300-EX platform switches have dedicated TCAM and do not require carving.
- TCAM carving configuration of the ing-netflow region can be performed on FX line cards. EX line cards have a default ing-netflow region TCAM carving of 1024 and cannot be configured otherwise. For ports on the EX and FX line cards, the suggested maximum for the ing-netflow region is 1024.
- The ToS field is not exported for Cisco Nexus 9300-EX platform switches.
- Record match that is based on IP ToS, is not supported for IPv6 flow monitors. The ToS value is collected on the collector as 0x0 irrespective of the value the traffic holds.

  This limitation is applicable for the following platform switch families:

  - Cisco Nexus 9300-EX

  - Cisco Nexus 9300-FX
  - Cisco Nexus 9300-FX2
  - Cisco Nexus 9500 with EX and FX line cards

-
- The Cisco Nexus 3232C and 3264Q switches do not support NetFlow.
- Enable NetFlow only on platforms that support this feature.

Note For verified NetFlow scalability numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

Configuring NetFlow

Follow these steps to configure NetFlow:

Procedure

Step 1 Enable the NetFlow feature.
Step 2 Define a flow record by specifying keys and fields to the flow.
Step 3 Define an optional flow exporter by specifying the export format, protocol, destination, and other parameters.
Step 4 Define a flow monitor based on the flow record and flow exporter.
Step 5 Apply the flow monitor to a source interface, subinterface, or VLAN interface.

Enabling the NetFlow Feature

You must globally enable NetFlow before you can configure any flows.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature netflow** **Example:** `switch(config)# feature netflow`	Enables or disables the NetFlow feature. The default is disabled.

	Command or Action	Purpose
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 a Flow Record

You can create a flow record and add keys to match on and nonkey fields to collect in the flow.

Procedure

	Command or Action	Purpose

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

Step 2	**flow record** *name*
	Example:
	`switch(config)# flow record Test`
	`switch(config-flow-record)#`
		Creates a flow record and enters flow record configuration mode. You can enter up to 63 alphanumeric characters for the flow record name.

Step 3	(Optional) **description** *string* **Example:** `switch(config-flow-record)# description IPv4Flow`	Describes this flow record as a maximum 63-character string.
Step 4	(Optional) **match** *type* **Example:** `switch(config-flow-record)# match transport destination-port`	Specifies a match key. For more information, see Specifying the Match Parameters, on page 290. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer 4 port data.
Step 5	(Optional) **collect** *type* **Example:** `switch(config-flow-record)# collect counter packets`	Specifies the collection field. For more information, see Specifying the Collect Parameters, on page 291.
Step 6	(Optional) **show flow record** [*name*] [*record-name*] {**netflow-original** | **netflow protocol-port** | **netflow** {**ipv4** | **ipv6**} {**original-input** | **original-output**}} **Example:**	Displays information about NetFlow flow records. You can enter up to 63 alphanumeric characters for the flow record name.

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

Specifying the Match Parameters

You must configure at least one of the following match parameters for flow records:

Command	Purpose
match datalink {**mac source-address** | **mac destination-address** | **ethertype** | **vlan**} **Example:** `switch(config-flow-record)# match datalink ethertype`	Specifies the Layer 2 attribute as a key.
match ip {**protocol** | **tos**} **Example:** `switch(config-flow-record)# match ip protocol`	Specifies the IP protocol or ToS fields as keys. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer 4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.
match ipv4 {**destination address** | **source address**} **Example:** `switch(config-flow-record)# match ipv4 destination address`	Specifies the IPv4 source or destination address as a key.
match ipv6 {**destination address** | **source address** | **flow-label** | **options**} **Example:** `switch(config-flow-record)# match ipv6 flow-label`	Specifies the IPv6 key.

Command	Purpose
match transport {**destination-port** | **source-port**} **Example:** `switch(config-flow-record)# match transport destination-port`	Specifies the transport source or destination port as a key. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer 4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.

Specifying the Collect Parameters

You must configure at least one of the following collect parameters for flow records:

Command	Purpose
collect counter {**bytes** | **packets**} [**long**] **Example:** `switch(config-flow-record)# collect counter packets`	Collects either packet-based or byte counters from the flow. You can optionally specify that 64-bit counters are used.
collect ip version **Example:** `switch(config-flow-record)# collect ip version`	Collects the IP version for the flow.
collect timestamp sys-uptime {**first** | **last**} **Example:** `switch(config-flow-record)# collect timestamp sys-uptime last`	Collects the system up time for the first or last packet in the flow.
collect transport tcp flags **Example:** `switch(config-flow-record)# collect transport tcp flags`	Collects the TCP transport layer flags for the packets in the flow.

Creating a Flow Exporter

The flow exporter configuration defines the export parameters for a flow and specifies reachability information for the remote NetFlow Collector.

Procedure

	Command or Action	Purpose

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

Step 2	**flow exporter** *name*
	Example:
	`switch(config)# flow exporter flow-exporter-one`
	`switch(config-flow-exporter)#`
		Creates a flow exporter and enters flow exporter configuration mode. You can enter up to 63 alphanumeric characters for the flow exporter name.

Step 3	**destination** {*ipv4-address* | *ipv6-address*} [**use-vrf** *name*] **Example:** `switch(config-flow-exporter)# destination 192.0.2.1`	Sets the destination IPv4 or IPv6 address for this flow exporter. You can optionally configure the VRF to use to reach the NetFlow Collector. You can enter up to 32 alphanumeric characters for the VRF name.
Step 4	**source** *interface-type name/port* **Example:** `switch(config-flow-exporter)# source ethernet 2/1`	Specifies the interface to use to reach the NetFlow Collector at the configured destination.
Step 5	(Optional) **description** *string* **Example:** `switch(config-flow-exporter)# description exportversion9`	Describes this flow exporter. You can enter up to 63 alphanumeric characters for the description.
Step 6	(Optional) **dscp** *value* **Example:** `switch(config-flow-exporter)# dscp 0`	Specifies the differentiated services codepoint value. The range is from 0 to 63.
Step 7	(Optional) **transport udp** *port* **Example:** `switch(config-flow-exporter)# transport udp 200`	Specifies the UDP port to use to reach the NetFlow Collector. The range is from 0 to 65535. **Note** If you do not specify the UDP port, 9995 is selected as the default.

Step 8	**version 9**
	Example:
	`switch(config-flow-exporter)# version 9`
	`switch(config-flow-exporter-version-9)#`
		Specifies the NetFlow export version. Choose version 9 to enter the flow exporter version 9 configuration submode.

	Command or Action	Purpose
Step 9	(Optional) **option** {**exporter-stats** | **interface-table**} **timeout** *seconds* **Example:** `switch(config-flow-exporter-version-9)# option exporter-stats timeout 1200`	Sets the flow exporter statistics resend timer. The range is from 1 to 86400 seconds.
Step 10	(Optional) **template data timeout** *seconds* **Example:** `switch(config-flow-exporter-version-9)# template data timeout 1200`	Sets the template data resend timer. The range is from 1 to 86400 seconds.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config-flow-exporter-version-9)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating a Flow Monitor

You can create a flow monitor and associate it with a flow record and a flow exporter. All of the flows that belong to a monitor use the associated flow record to match on the different fields, and the data is exported to the specified flow exporter.

Procedure

	Command or Action	Purpose

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

Step 2	**flow monitor** *name*
	Example:
	`switch(config)# flow monitor flow-monitor-one`
	`switch(config-flow-monitor)#`
		Creates a flow monitor and enters flow monitor configuration mode. You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 3	(Optional) **description** *string* **Example:** `switch(config-flow-monitor)# description IPv4Monitor`	Describes this flow monitor. You can enter up to 63 alphanumeric characters for the description.
Step 4	(Optional) **exporter** *name* **Example:** `switch(config-flow-monitor)# export v9`	Associates a flow exporter with this flow monitor. You can enter up to 63 alphanumeric characters for the exporter name.

	Command or Action	Purpose
Step 5	**record** *name* [**netflow-original** | **netflow protocol-port** | **netflow** {**ipv4** | **ipv6**} {**original-input** | **original-output**}] **Example:** `switch(config-flow-monitor)# record IPv4Flow`	Associates a flow record with the specified flow monitor. You can enter up to 63 alphanumeric characters for the record name.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-flow-monitor)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Applying a Flow Monitor to an Interface

You can apply a flow monitor to an ingress interface. Egress Netflow is not supported.

Procedure

	Command or Action	Purpose

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

Step 2	**interface vlan** *vlan-id*
	Example:
	`switch(config)# interface vlan 10`
	`switch(config-if)#`
		Configures a VLAN interface and enters interface configuration mode.

Step 3	**ip flow monitor** {**ipv4** | **ipv6** | **layer-2-switched**} **input** **Example:** `switch(config-if)# ip flow monitor ipv4 input`	Associates an IPv4, IPv6, or Layer 2-switched flow monitor to the interface for input packets.
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.

Configuring Bridged NetFlow on a VLAN

You can apply a flow monitor to a VLAN in order to gather Layer 3 data over Layer 2 switched packets in a VLAN.

Procedure

	Command or Action	Purpose

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

Step 2	**vlan configuration** *vlan-id*
	Example:
	`switch(config)# vlan configuration 30`
	`switch(config-vlan-config)#`
		Enters VLAN configuration mode. The VLAN ID range is from 1 to 3967 or from 4048 to 4093.
		Note VLAN configuration mode enables you to configure VLANs independently of their creation, which is required for VTP client support.

Step 3	{**ip** | **ipv6**} **flow monitor** *name* **Example:** `switch(config-vlan-config)# ip flow monitor testmonitor`	Associates a flow monitor to the VLAN for input packets. You can enter up to 63 alphanumeric characters for the flow monitor name.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-vlan-config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Layer 2 NetFlow Keys

You can define Layer 2 keys in flexible NetFlow records that you can use to capture flows in Layer 2 interfaces.

Procedure

	Command or Action	Purpose

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

Step 2	**flow record** *name*
	Example:
	`switch(config)# flow record L2_record`
	`switch(config-flow-record)#`
		Enters flow record configuration mode. For more information about configuring flow records, see Creating a Flow Record, on page 289.

Step 3	**match datalink** {**mac source-address** | **mac destination-address** | **ethertype** | **vlan**} **Example:**	Specifies the Layer 2 attribute as a key.

	Command or Action	Purpose
	`switch(config-flow-record)# match datalink ethertype`	

Step 4	**exit**
	Example:
	`switch(config-flow-record)# exit`
	`switch(config)#`
		Exits flow record configuration mode.

Step 5	**interface** {**ethernet** *slot/port* | **port-channel** *number*}
	Example:
	`switch(config)# interface Ethernet 6/3`
	`switch(config-if#)`
		Enters interface configuration mode. The interface type can be a physical Ethernet port or a port channel.

Step 6	**switchport** **Example:** `switch(config-if)# switchport`	Changes the interface to a Layer 2 physical interface. For information on configuring switch ports, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.
Step 7	**mac packet-classify** **Example:** `switch(config-if)# mac packet-classify`	Forces MAC classification of packets. For more information on using this command, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. **Note** You must use this command to capture flows.
Step 8	**layer2-switched flow monitor** *flow-name* **input** **Example:** `switch(config-if)# layer2-switched flow monitor L2_monitor input`	Associates a flow monitor to the switch port input packets. You can enter up to 63 alphanumeric characters for the flow monitor name.
Step 9	(Optional) **show flow record netflow layer2-switched input** **Example:** `switch(config-if)# show flow record netflow layer2-switched input`	Displays information about the Layer 2 NetFlow default record.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring NetFlow Timeouts

You can optionally configure global NetFlow timeouts that apply to all flows in the system.

Procedure

	Command or Action	Purpose

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

Step 2	**flow timeout** *seconds* **Example:** `switch(config)# flow timeout 30`	Sets the flush timeout value in seconds. The range is from 5 to 60 seconds. The default value is 10 seconds.
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 the NetFlow Configuration

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

Command	Purpose
show flow cache [**ipv4** | **ipv6** | **ce**]	Displays information about NetFlow IP flows.
show flow exporter [*name*]	Displays information about NetFlow flow exporters and statistics. You can enter up to 63 alphanumeric characters for the flow exporter name.
show flow interface [*interface-type slot/port*]	Displays information about NetFlow interfaces.
show flow record [*name*]	Displays information about NetFlow flow records. You can enter up to 63 alphanumeric characters for the flow record name.
show flow record netflow layer2-switched input	Displays information about the Layer 2 NetFlow configuration.
show running-config netflow	Displays the NetFlow configuration that is currently on your device.

Monitoring NetFlow

Use the **show flow exporter** command to display NetFlow statistics. Use the **clear flow exporter** command to clear NetFlow flow exporter statistics.

Configuration Example for NetFlow

This example shows how to configure a NetFlow exporter configuration for IPv4:

```
feature netflow
flow exporter ee
 destination 171.70.242.48 use-vrf management
 source mgmt0
 version 9
  template data timeout 20
flow record rr
 match ipv4 source address
 match ipv4 destination address
 collect counter bytes
 collect counter packets
flow monitor foo
 record rr
 exporter ee
interface Ethernet2/45
 ip flow monitor foo input
 ip address 10.20.1.1/24
 no shutdown
```

CHAPTER 20

Configuring sFlow

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

This chapter includes the following sections:

About sFlow

Sampled flow (sFlow) allows you to monitor real-time traffic in data networks that contain switches and routers. It uses the sampling mechanism in the sFlow agent software on switches and routers to monitor traffic and to forward the sample data to the central data collector.

For more information about sFlow, see RFC 3176.

sFlow Agent

The sFlow agent, which is embedded in the Cisco NX-OS software, periodically samples or polls the interface counters that are associated with a data source of the sampled packets. The data source can be an Ethernet interface, an EtherChannel interface, or a range of Ethernet interfaces. The sFlow agent queries the Ethernet port manager for the respective EtherChannel membership information and also receives notifications from the Ethernet port manager for membership changes.

When you enable sFlow sampling, based on the sampling rate and the hardware internal random number, the ingress packets and egress packets are sent to the CPU as an sFlow-sampled packet. The sFlow agent processes the sampled packets and sends an sFlow datagram to the sFlow analyzer. In addition to the original sampled packet, an sFlow datagram includes information about the ingress port, the egress port, and the original packet length. An sFlow datagram can have multiple sFlow samples.

Prerequisites for sFlow

sFlow has the following prerequisites:

- For Cisco Nexus 9332PQ, 9372PX, 9372TX, and 93120TX switches and for Cisco Nexus 9396PX, 9396TX, and 93128TX switches with the N9K-M6PQ generic expansion module (GEM), you must configure the sFlow and SPAN ACL TCAM region sizes for any uplink ports that are to be configured as an sFlow data source. To do so, use the **hardware access-list tcam region sflow** and **hardware access-list tcam region span** commands. See Configuring ACL TCAM Region Sizes for more information.

Note By default, the sflow region size is zero, and the span region size is non-zero. You need to configure the sflow region to 256 and allocate enough entries to the span region in order to configure the port as an sFlow data source.

- Egress sFlow of multicast traffic requires **hardware multicast global-tx-span** configuration

Guidelines and Limitations for sFlow

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

sFlow has the following guidelines and limitations:

- If at least one sFlow data source is configured, the SPAN sessions cannot be brought up.
  - If at least one SPAN session is configured as **no shut** , sFlow data sources cannot be added.
  - The sampling mode that is used for sFlow is based on an algorithm that is known as LFSR. Due to the use of LFSR, it is not guaranteed that one in every few packets are sampled with the sampling rate of n. However, the number of packets that are sampled is equal to the total packets over a period of time.
- When sFlow and SPAN are configured on the same interface, and the hardware rate-limiter is configured for sFlow, the Rate-Limiter Drops counter in the output of the **show hardware rate-limiter** command displays more drops than expected.
- sFlow is a software-driven feature, hardware only sends copies of traffic from the sFlow source interfaces to the CPU for further processing. Elevated CPU usage is expected. sFlow traffic sent to the CPU by hardware is rate-limited to protect the CPU.
- When you enable sFlow for an interface, it is enabled for both ingress and egress. You cannot enable sFlow for only ingress or only egress.

  For Cisco Nexus 9508 switches with Cisco Nexus 9636C-R and 9636Q-R line cards, sFlow can be enabled for an interface only in the ingress direction.
- The storm control feature does not work if you enable storm control on an interface where sFlow is also enabled.

- sFlow is not supported on the SVIs.
- Subinterfaces are not supported for sFlow.
- We recommend you configure the sampling rate that is based on the sFlow configuration and traffic in the system.
- The switch supports only one sFlow collector.
- sFlow and Network Address Translation (NAT) are not supported on the same port.
- sFlow does not support egress sampling for multicast, broadcast, or unknown unicast packets.

  sFlow supports sampling IPv6 traffic but only on IPv4 collector address.
- sFlow counters increment even for control packets that ingress on the sFlow data-source interfaces. These packets may be sampled and send out as sFlow datagrams (similar to data plane traffic).
- Nexus 9000-EX, FX, and GX family of switches only support sampling at the following values: 4096, 8192, 16384, 32768, 65536. Configuring values other than these results in the value being rounded off to the next supported value.

Default Settings for sFlow

The following table lists the default settings for sFlow parameters.

Table 21: Default sFlow Parameters

Parameters	Default
sFlow sampling rate	4096
sFlow sampling size	128
sFlow counter poll interval	20
sFlow maximum datagram size	1400
sFlow collector IP address	0.0.0.0
sFlow collector port	6343
sFlow agent IP address	0.0.0.0

Configuring sFlow

Enabling sFlow

You must enable the sFlow feature before you can configure sFlow settings on the switch.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature sflow** **Example:** `switch(config)# feature sflow`	Enables or disables sFlow.
Step 3	(Optional) **show feature** **Example:** `switch(config)# show feature`	Displays the enabled and disabled features.
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 the Sampling Rate

You can configure the sampling rate for sFlow.

Before you begin

Make sure that you have enabled sFlow.

Nexus 9000-EX, FX, and GX family of switches only support sampling at the following values: 4096, 8192, 16384, 32768, 65536. Configuring values other than these will result in the value being rounded off to the next supported value.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow sampling-rate** *sampling-rate* **Example:** `switch(config)# sflow sampling-rate 50000`	Configures the sFlow sampling rate for packets. The *sampling-rate* can be an integer between 4096 and 1000000000.
Step 3	(Optional) **show sflow** **Example:**	Displays the sFlow configuration.

	Command or Action	Purpose
	`switch(config)# show sflow`	
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 the Maximum Sampled Size

You can configure the maximum number of bytes that should be copied from a sampled packet.

Before you begin

Make sure that you have enabled sFlow.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow max-sampled-size** *sampling-size* **Example:** `switch(config)# sflow max-sampled-size 200`	Configures the sFlow maximum sampling size. The range for the *sampling-size* is from 64 to 256 bytes.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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 the Counter Poll Interval

You can configure the maximum number of seconds between successive samples of the counters that are associated with the data source. A sampling interval of 0 disables counter sampling.

Before you begin

Make sure that you have enabled sFlow.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow counter-poll-interval** *poll-interval* **Example:** `switch(config)# sflow counter-poll-interval 100`	Configures the sFlow poll interval for an interface. The range for the *poll-interval* is from 0 to 2147483647 seconds.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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 the Maximum Datagram Size

You can configure the maximum number of data bytes that can be sent in a single sample datagram.

Before you begin

Make sure that you have enabled sFlow.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow max-datagram-size** *datagram-size* **Example:** `switch(config)# sflow max-datagram-size 2000`	Configures the sFlow maximum datagram size. The range for the *datagram-size* is from 200 to 9000 bytes.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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 the sFlow Collector Address

You can configure the IPv4 address of the sFlow data collector that is connected to the management port.

Before you begin

Make sure that you have enabled sFlow.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow collector-ip** *ip-address* **vrf** *vrf* [**source** *ip-address*] **Example:** `switch(config)# sflow collector-ip 192.0.2.5 vrf management`	Configures the IPv4 address for the sFlow collector. If the IP address is set to 0.0.0.0, all samples will be dropped. The *vrf* can be one of the following: • A user-defined VRF name—You can specify a maximum of 32 alphanumeric characters. • **vrf management**—You must use this option if the sFlow data collector is on the network connected to the management port. • **vrf default**—You must use this option if the sFlow data collector is on the network connected to the front-panel ports. The **source** *ip-address* option causes the sent sFlow datagram to use the source IP address as the IP packet source address. The source IP address has to be already configured on one of the switch local interfaces; otherwise, an error message appears. If the interface with the source IP address is changed or removed after this option is configured, the sFlow datagram will no longer be sent out, and an event history error

	Command or Action	Purpose
		and syslog error will be logged. When the **source** *ip-address* option is not configured, Cisco NX-OS picks the IP packet source address automatically for the sent sFlow datagram.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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 the sFlow Collector Port

You can configure the destination port for sFlow datagrams.

Before you begin

Make sure that you have enabled sFlow.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow collector-port** *collector-port* **Example:** `switch(config)# sflow collector-port 7000`	Configures the UDP port of the sFlow collector. The range for the *collector-port* is from 1 to 65535.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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 the sFlow Agent Address

You can configure the IPv4 address of the sFlow agent.

Before you begin

Make sure that you have enabled sFlow.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow agent-ip** *ip-address* **Example:** `switch(config)# sflow agent-ip 192.0.2.3`	Configures the IPv4 address of the sFlow agent. The default IP address is 0.0.0.0, which means that all samples will be dropped. You must specify a valid IP address to enable sFlow functionality. **Note** This IP address is not necessarily the source IP address for sending the sFlow datagram to the collector.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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 the sFlow Sampling Data Source

You can configure the source of the data for the sFlow sampler as an Ethernet port, a range of Ethernet ports, or a port channel.

Before you begin

Make sure that you have enabled sFlow.

If you want to use a port channel as the data source, make sure that you have already configured the port channel and you know the port channel number.

Make sure that the sFlow and SPAN ACL TCAM region sizes are configured for any uplink ports that are to be configured as an sFlow data source on the following devices: Cisco Nexus 9332PQ, 9372PX, 9372TX, and 93120TX switches and Cisco Nexus 9396PX, 9396TX, and 93128TX switches with the N9K-M6PQ or N9K-M12PQ generic expansion module (GEM).

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **sflow data-source interface** [**ethernet** *slot*/*port*[-*port*] | **port-channel** *channel-number*] **Example:** `switch(config)# sflow data-source interface ethernet 1/5-12`	Configures the sFlow sampling data source. For an Ethernet data source, *slot* is the slot number, and *port* can be either a single port number or a range of ports designated as *port-port*.
Step 3	(Optional) **show sflow** **Example:** `switch(config)# show sflow`	Displays the sFlow 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.

Verifying the sFlow Configuration

Use these commands to display the sFlow configuration.

Table 22: sFlow Show Commands

Command	Purpose
show sflow	Displays all the data sources of the sFlow samplers and the sFlow agent configuration.
show process	Verifies whether the sFlow process is running.
show running-config sflow [**all**]	Displays the current sFlow running configuration.

Monitoring and Clearing sFlow Statistics

Use the **show sflow statistics** command to display the sFlow statistics.

Use the following commands to clear the sFlow statistics:

Command	Description
clear sflow statistics	Clears most of the sFlow statistics from the **show sflow statistics** command.
clear counters interface all	Clears the Total Packets field from the **show sflow statistics** command.
clear hardware rate-limiter sflow	Clears the Total Samples field from the **show sflow statistics** command.

Configuration Examples for sFlow

This example shows how to configure sFlow:

```
feature sflow
sflow sampling-rate 4096
sflow max-sampled-size 200
sflow counter-poll-interval 100
sflow max-datagram-size 2000
sflow collector-ip 192.0.2.5 vrf management
sflow collector-port 7000
sflow agent-ip 192.0.2.3
sflow data-source interface ethernet 1/5
```

Additional References

Related Documents

Related Topic	Document Title
ACL TCAM regions	Configuring IP ACLs

CHAPTER 21

Configuring TAP Aggregation and MPLS Stripping

This chapter describes how to configure TAP aggregation and MPLS stripping on Cisco NX-OS devices.

This chapter contains the following sections:

About TAP Aggregation

Network TAPs

You can use various methods to monitor packets. One method uses physical hardware test access points (TAPs).

Network TAPs can be extremely useful in monitoring traffic because they provide direct inline access to data that flows through the network. In many cases, a third party monitors the traffic between two points in the network. If the network between points A and B consists of a physical cable, a network TAP might be the best way to accomplish this monitoring. The network TAP has at least three ports: an A port, a B port, and a monitor port. A TAP inserted between the A and B ports passes all traffic through unimpeded, but it also copies that same data to its monitor port, which could enable a third party to listen.

TAPs have the following benefits:

- They can handle full-duplex data transmission.
- They are unobtrusive and not detectable by the network (with no physical or logical addressing).
- Some TAPs support full inline power with the capability to build a distributed TAP.

If you are trying to gain visibility into the server-to-server data communication at the edge or virtual edge of your network or to provide a copy of traffic to the Intrusion Prevention System (IPS) appliance at the Internet edge of your network, you can use network TAPs nearly anywhere in the environment. However, this deployment can add significant costs, operation complexities, and cabling challenges in a large-scale environment.

TAP Aggregation

TAP aggregation is an alternative solution to help with monitoring and troubleshooting tasks in the data center. It works by designating a device to allow the aggregation of multiple test access points (TAPs) and to connect to multiple monitoring systems. TAP aggregation switches link all of the monitoring devices to specific points in the network fabric that handle the packets that need to be observed.

In the TAP aggregation switch solution, a Cisco Nexus 9000 Series switch is connected to various points in the network at which packet monitoring is advantageous. From each network element, you can use switched port analyzer (SPAN) ports or optical TAPs to send traffic flows directly to this TAP aggregation switch. The TAP aggregation switch is directly connected to all of the analysis tools used to monitor the events in the network fabric. These monitoring devices include remote monitor (RMON) probes, application firewalls, IPS devices, and packet sniffer tools.

You can configure the TAP aggregation switch to filter specific traffic and redirect it to one or more tools. In order to redirect the traffic to multiple interfaces, a multicast group is created internally on the switch, and the interfaces that are part of the redirect list are added as member ports. When an access control list (ACL) policy with the redirect action is applied to an interface, the traffic matching the ACL rule is redirected to the internal multicast group that is created.

Guidelines and Limitations for TAP Aggregation

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

TAP aggregation has the following guidelines and limitations:

- TAP aggregation:
  - Supported on all Cisco Nexus 9000 Series switches and the 3164Q, 31128PQ, 3232C, and 3264Q switches.
  - Supported on 100G ports.
  - Supports only on switch ports and only in the ingress direction.
  - Supports IPv4 ACLs with UDF-based match for Cisco Nexus 9200, 9300, and 9300-EX Series switches.
  - Supported on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9500-EX, and 9500-FX platform switches.
  - Maximum redirect ports supported are 32 interfaces.
- Beginning with Cisco NX-OS Release 9.2(1), TAP aggregation filters on MPLS tags are supported on the following Cisco Nexus platform switches:

- Cisco Nexus 9000 platform switches, including the 9700-EX and 9700-FX line cards.
- Cisco Nexus 9200 platform switches.
- Cisco Nexus 9300 platform switches.
- Cisco Nexus 9500 switches.

- TAP aggregation filters on MPLS tags are not supported on the following Cisco Nexus Series switches, line cards, and fabric modules:

Table 23: Cisco Nexus 9000 Series Switches

Cisco Nexus 3164Q-40GE	Cisco Nexus 9372PX	Cisco Nexus 9372PX-E
Cisco Nexus 9372TX	Cisco Nexus 9372TX-E	Cisco Nexus 9332PQ
Cisco Nexus 3232C	Cisco Nexus 93120TX	Cisco Nexus 31128PQ
Cisco Nexus 3264Q-S	—	—

Table 24: Cisco Nexus 9000 Series Line Cards and Fabric Modules

N9K-X9564PX	N9K-X9564TX	N9K-X9536PQ
N9K-M12PQ	N9K-C93128TX	N9K-C9396PX
N9K-C9396TX	N9K-X9408PC-CFP2	N9K-M4PC-CFP2
N9K-X9432PQ	N9K-X9464PX	N9K-X9464TX
N9K-X9464TX2	N9K-X9632PC-QSFP100	N9K-X9432C-S
N9K-M6PQ	N9K-M6PQ-E	—

- Cisco Nexus 9700-EX and 9700-FX line cards support TAP aggregation with IPv4, IPv6, and MAC ACLs.
- Only Layer 2 interfaces support the TAP aggregation policy. You can apply the policy to a Layer 3 interface, but the policy becomes nonfunctional.
- The redirect port must be part of the same VLAN as the source (TAP) port.
- Each rule must be associated with only one unique match criterion.
- When you enter a list of interfaces for the TAP aggregation policy, you must separate them with commas but no spaces. For example, port-channel50, ethernet1/12, port-channel20.
- When you specify target interfaces in a policy, make sure that you enter the whole interface type and not an abbreviated version. For example, make sure that you enter **ethernet1/1** instead of **eth1/1** and **port-channel50** instead of **po50**.
- HTTP requests with *tcp-option-length* and *VLAN ID* filters simultaneously are not supported. Traffic match against ACE may not work if you configure both filters at a time.

- When configuring ACL entries with redirect to port-channels that are yet to be configured, the user must take care to configure the specified port-channels at a later point of time.

About MPLS Stripping

The ingress ports of Cisco Nexus 9000 Series switches receive various Multiprotocol Label Switching (MPLS) packet types. Each data packet in an MPLS network has one or more label headers. These packets are redirected on the basis of a redirect access control list (ACL).

A label is a short, four-byte, fixed-length, locally significant identifier that is used to identify a Forwarding Equivalence Class (FEC). The label that is put on a particular packet represents the FEC to which that packet is assigned. It has the following components:

- Label—Label value (unstructured), 20 bits
- Exp—Experimental use, 3 bits; currently used as a class of service (CoS) field
- S—Bottom of stack, 1 bit
- TTL—Time to live, 8 bits

Some MPLS labels are imposed between the Layer 2 header and the Layer 3 header. For these labels, the headers and data are not located at the standard byte offset. Standard network monitoring devices cannot monitor and analyze this traffic. single-labeled packets are stripped off their MPLS label headers and redirected to T-cache devices.

MPLS packets with multiple label headers are sent to deep packet inspection (DPI) devices without stripping their MPLS headers.

Beginning Cisco NX-OS Release 7.0(3)I7(3), for Cisco Nexus 9300- EX switches, a VLAN tag can be applied to packets going out of redirect ports. You can pop 1-5 labels with the MPLS strip.

Guidelines and Limitations for MPLS Stripping

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

MPLS stripping has the following guidelines and limitations:

- Cisco Nexus 9700-EX and 9700-FX line cards do not support MPLS stripping.
- Cisco Nexus 9000-FX2 switches (N9K-C93216TC-FX2, N9K-C93240YC-FX2, N9K-C93360YC-FX2 and N9K-C9336C-FX2) do not support the MPLS stripping for EoMPLS encapsulation type. However, the IPoMPLS strip is supported.
- Disable all Layer 3 and vPC features before you enable MPLS stripping.
- Static MPLS, MPLS segment routing, and MPLS stripping cannot be enabled at the same time.
- Only the ingress interfaces involved in MPLS stripping must have TAP aggregation enabled.
- You must configure the TAP aggregation ACL with a redirect action on the ingress interface to forward the packet to the desired destination.

- Only one TAP ACL is supported on the system.
- Post MPLS strip, SMAC changes to switch mac (**show vdc**) and DMAC is set to **00:00:00:ab:cd:ef**.
- The egress interface where stripped packets will exit must be an interface that has VLAN 1 as an allowed VLAN. We recommend that you configure the egress interface as a trunk with all VLANs allowed by default.
- Stripping is based on IP PACL, and you cannot use MAC-ACL for stripping.
- MPLS stripping is supported only for IPv4 traffic.
- Port-channel load balancing is supported for MPLS stripped packets.
- Layer 3 header-based hashing and Layer 4 header-based hashing are supported, but Layer 2 header-based hashing is not supported.
- During MPLS stripping, the incoming VLAN is not preserved.
- Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches support tagging of VLANs to packets going out of redirect ports. The ingress/egress ports can either be ethernet or port channel. The VLAN tag is derived from the incoming port configuration. The new ACL on the ingress interface should not be associated with a VLAN value different from the interface VLAN value.
- For every ACE (under an ACL associated with a particular VLAN) with a unique redirect port list, we allocate a hardware entry. The current hardware limit for the number of ACEs is 50 and you cannot configure more than 50 such ACEs.
- MPLS strip is only supported for Layer 3 packets under the MPLS label stack.
- MPLS strip is not supported for pseudowires or VPLS.

Configuring TAP Aggregation

Enabling TAP Aggregation for Line Cards

Beginning with Cisco NX-OS Release 7.0(3)I7(2), you can enable TAP aggregation for Cisco Nexus 9500 platform switches with 9700-EX and 9700-FX line cards.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **hardware acl tap-agg** **Example:**	Enables TAP aggregation for Cisco Nexus 9700-EX and 9700-FX line cards.

	Command or Action	Purpose
	`switch(config)# hardware acl tap-agg`	This command is also needed on Cisco Nexus 9300-GX platform switches and may require reload.
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 a TAP Aggregation Policy

You can configure a TAP aggregation policy on an IP access control list (ACL) or on a MAC ACL.

Before you begin

You must configure the ACL TCAM region size for IPv4 port ACLs or MAC port ACLs using the **hardware access-list tcam region** {*ifacl* | *mac-ifacl*} command. Configure the ACL TCAM region size for IPv6 port ACLs using the command, **hardware access-list team region ipv6-ifcal**.

For information, see the "Configuring ACL TCAM Region Sizes" in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Note By default the region size for both ifacl and mac-ifacl is zero. You need to allocate enough entries to the ifacl or mac-ifacl region to support TAP aggregation.

Procedure

	Command or Action	Purpose

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

Step 2	Enter one of the following commands:
	• **ip access-list** *access-list-name*
	• **mac access-list** *access-list-name*
	Example:
	`switch(config)# ip access-list test`
	`switch(config-acl)#`
	
	`switch(config)# mac access-list mactap1`
	`switch(config-mac-acl)#`
	
	`switch(config)# ipv6 access-list testv6`
	
	`switch# sh running-config aclmgr`
		Note Beginning with Cisco NX-OS Release 7.0(3)I5(1), support for IPv6 ACLs is added for Cisco Nexus 9000 Series switches. The redirect action is supported in IPv6 ACLs. All the match options that are currently supported for IPv6 PACLs are now supported with the redirect action.
		
		Creates an IPACL and enters IP access list configuration mode or creates a MAC ACL and enters MAC access list configuration mode.

	Command or Action	Purpose

	`!Command: show running-config aclmgr`
	`!Time: Thu Aug 11 18:19:50 2016`
	
	`version 7.0(3)I5(1)`
	`ipv6 access-list testv6`
	`10 permit ipv6 any any redirect Ethernet2/1`
	
	`interface Ethernet6/6`
	`ipv6 port traffic-filter testv6 in`	

Step 3	(Optional) **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Starts recording statistics for how many packets are permitted or denied by each entry.
Step 4	[**no**] **permit** *protocol source destination* **redirect** *interfaces* **Example:** `switch(config-acl)# permit ip any any redirect ethernet1/8`	Creates an IP or MAC ACL rule that permits traffic to be redirected per its conditions. The **no** version of this command removes the permit rule from the policy. **Note** When you enter an interface for the TAP aggregation policy, do not abbreviate it. When you enter a list of interfaces, separate them with commas but no spaces.

Step 5	(Optional) Enter one of the following commands:
	• **show ip access-lists** [*access-list-name*]
	• **show mac access-lists** [*access-list-name*]
	Example:
	`switch(config-acl)# show ip access-lists test`
	
	`switch(config-mac-acl)# show mac access-lists mactap1`
		Displays all IPv4 or MAC ACLs or a specific IPv4 or MAC ACL.

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

Attaching a TAP Aggregation Policy to an Interface

You can apply an ACL configured with TAP aggregation to a Layer 2 interface.

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`
	`switch(config-if)#`
		Enters interface configuration mode for the specified interface.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Changes a Layer 3 interface to a Layer 2 interface. **Note** Make sure that the interface is a Layer 2 interface.

Step 4	Enter one of the following commands:
	• [**no**] **ip port access-group** *access-list-name* **in**
	• [**no**] **mac port access-group** *access-list-name* **in**
	Example:
	`switch(config-if)# ip port access-group test in`
	`switch(config-if)# mac port access-group test in`
		Applies an IPv4 or MAC ACL configured with TAP aggregation to the interface. The **no** form of this command removes the ACL from the interface.

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

Verifying the TAP Aggregation Configuration

To display the TAP aggregation configuration information, perform one of the following tasks.

Command	Purpose
show ip access-lists [*access-list-name*]	Displays all IPv4 ACLs or a specific IPv4 ACL.
show mac access-lists [*access-list-name*]	Displays all MAC ACLs or a specific MAC ACL.

Configuration Example for TAP Aggregation

This example shows how to configure a TAP aggregation policy on an IPv4 ACL:

```
switch# configure terminal

switch(config)# ip access-list test
switch(config-acl)# 10 deny ip 100.1.1/24 any
switch(config-acl)# 20 permit tcp any eq www any redirect port-channel4
switch(config-acl)# 30 permit ip any any redirect
Ethernet1/1,Ethernet1/2,port-channel7,port-channel8,Ethernet1/12,Ethernet1/13
switch(config-acl)# show ip access-lists test
IP access list test
        10 deny ip 100.1.1/24 any
        20 permit tcp any eq www any redirect port-channel4
        30 permit ip any any redirect
Ethernet1/1,Ethernet1/2,port-channel7,port-channel8,Ethernet1/12,Ethernet1/13
```

This example shows how to apply a TAP aggregation policy to an IPv4 ACL with UDF–based match.

```
switch# configure terminal

switch(config)# ip access-list tap_agg
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 permit ip any any redirect Ethernet1/1
switch(config-acl)# 20 deny ip any any
switch# configure terminal
switch(config)# interface Ethernet1/4
switch(config-if)# ip port access-group tap_agg in
switch(config-if)# switchport
switch(config-if)# no shutdown
```

This example shows how to configure a TAP aggregation policy on a MAC ACL:

```
switch# configure terminal

switch(config)# mac access-list mactap1
switch(config-mac-acl)# 10 permit any any 0x86dd redirect port-channel1
switch(config-mac-acl)# show mac access-lists mactap1
MAC access list mactap1
        10 permit any any 0x86dd redirect port-channel1
```

This example shows how to attach a TAP aggregation policy to a Layer 2 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip port access-group test in
switch(config-if)#
```

Configuring MPLS Stripping

Enabling MPLS Stripping

You can enable MPLS stripping globally.

Before you begin

Disable all Layer 3 and vPC features before you enable MPLS stripping.

Attach an ACL with the tap aggregation policy to the Layer 2 interface or port channel using the **mode tap-aggregation** command. For more information, see Attaching a TAP Aggregation Policy to an Interface, on page 317.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **mpls strip** **Example:** `switch(config)# mpls strip`	Globally enables MPLS stripping. The **no** form of this command disables MPLS stripping.
Step 3	[**no**] **mpls strip mode dot1q** **Example:** `switch(config)# mpls strip mode dot1q`	Enables VLAN tagging on the packets coming from the redirect port. The VLAN that needs to be tagged must be specified in the ingress port.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Incoming Port for the VLAN Tag

The VLAN tag is derived from the incoming port configuration. The ingress/egress ports can either be ethernet or port channel.

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 1/26`
	`switch(config-if)#`
		Enters interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Changes a Layer 3 interface to a Layer 2 interface. **Note** Make sure that the interface is a Layer 2 interface.

Step 4	Enter one of the following commands:
	• [**no**] **ip port access-group** *access-list-name* **in**
	• [**no**] **mac port access-group** *access-list-name* **in**
	Example:
	`switch(config-if)# ip port access-group test in`
	`switch(config-if)# mac port access-group test in`
		Applies an IPv4 or MAC ACL configured with TAP aggregation to the interface. The **no** form of this command removes the ACL from the interface.

Step 5	Enter one of the following commands:
	• [**no**] **ip port access-group** *access-list-name* **in**
	• [**no**] **mac port access-group** *access-list-name* **in**
	Example:
	`switch(config-if)# ip port access-group test in`
	`switch(config-if)# mac port access-group test in`
		Applies an IPv4 or MAC ACL configured with TAP aggregation to the interface. The **no** form of this command removes the ACL from the interface.

Step 6	Required: **mode tap-aggregation vlan** *vlan_id* **Example:** `switch(config-if)# mode tap-aggregation vlan 26`	Configures the value of the VLAN that will be tagged to the packets coming from the redirect ports.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Adding and Deleting MPLS Labels

The device can learn the labels dynamically whenever a frame is received with an unknown label on a TAP interface. You can also add or delete static MPLS labels.

Before you begin

Configure a TAP aggregation policy and attach the policy to an interface. For more information, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

You must configure the TAP aggregation ACL with a redirect action on the ingress interface to forward the packet to the desired destination.

Procedure

	Command or Action	Purpose

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

Step 2	**mpls strip label** *label* **Example:** `switch(config)# mpls strip label 100`	Adds the specified static MPLS label. The 20-bit value of the label can range from 1 to 1048575. **Note** This CLI is available for all the platform switches specified for the MPLS Stripping feature in the Guidelines and Limitations section, except for the following cloud scale platform switches: • N9K-C93180YC-EX • N9K-C93180YC-FX • N9K-C93240YC-FX2 • N9K-C93180YC-FX3S The [**no**] **mpls strip label** {*label* | **all**} command deletes the specified static MPLS label. The **all** option deletes all static MPLS labels.
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 Destination MAC Addresses

You can configure the destination MAC address for stripped egress frames.

Procedure

	Command or Action	Purpose

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

Step 2	**mpls strip dest-mac** *mac-address* **Example:** `switch(config)# mpls strip dest-mac 1.1.1`	Specifies the destination MAC address for egress frames that are stripped of their headers. The MAC address can be specified in one of the following four formats: • E.E.E • EE-EE-EE-EE-EE-EE • EE:EE:EE:EE:EE:EE • EEEE.EEEE.EEEE
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 MPLS Label Aging

You can define the amount of time after which dynamic MPLS labels will age out, if unused.

Procedure

	Command or Action	Purpose

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

Step 2	**mpls strip label-age** *age* **Example:** `switch(config)# mpls strip label-age 300`	Specifies the amount of time in seconds after which dynamic MPLS labels age out. The range is from 61 to 31622400.
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 the MPLS Stripping Configuration

To display the MPLS stripping configuration, perform one of the following tasks:

Command	Purpose
show mpls strip labels [*label* | **all** | **dynamic** | **static**]	Displays information about MPLS labels. You can specify the following options: • *label*—Label to be displayed. • **all**—Specifies that all labels must be displayed. This is the default option. • **dynamic**—Specifies that only dynamic labels must be displayed. • **static**—Specifies that only static labels must be displayed.

This example shows how to display all MPLS labels:

```
switch# show mpls strip labels
MPLS Strip Labels:
    Total      : 3005
    Static     : 5
Legend:     * - Static Label
    Interface - where label was first learned
    Idle-Age  - Seconds since last use
    SW-Counter- Packets received in Software
    HW-Counter- Packets switched in Hardware
--------------------------------------------------------------------------------
     Label    Interface      Idle-Age   SW-Counter       HW-Counter
--------------------------------------------------------------------------------
     4096    Eth1/53/1          15          1              210
     4097    Eth1/53/1          15          1              210
     4098    Eth1/53/1          15          1              210
     4099    Eth1/53/1           7          2              219
     4100    Eth1/53/1           7          2              219
     4101    Eth1/53/1           7          2              219
     4102    Eth1/53/1          39          1              206
     4103    Eth1/53/1          39          1              206
     4104    Eth1/53/1          39          1              206
     4105    Eth1/53/1           1          1              217
     4106    Eth1/53/1           1          1              217
     4107    Eth1/53/1           1          1              217
     4108    Eth1/53/1          15          1              210
*   25000    None <User>        39          1              206
*   20000    None <User>        39          1              206
*   21000    None <User>         1          1              217
```

This example shows how to display only static MPLS labels:

```
switch(config)# show mpls strip labels static
MPLS Strip Labels:
    Total      : 3005
    Static     : 5
Legend:     * - Static Label
    Interface - where label was first learned
```

```
    Idle-Age  - Seconds since last use
    SW-Counter- Packets received in Software
    HW-Counter- Packets switched in Hardware
--------------------------------------------------------------------------------
     Label    Interface      Idle-Age   SW-Counter       HW-Counter
--------------------------------------------------------------------------------
*      300    None <User>        403          0                0
*      100    None <User>        416          0                0
*    25000    None <User>        869          0                0
*    20000    None <User>        869          0                0
*    21000    None <User>        869          0                0
```

Clearing MPLS Stripping Counters and Label Entries

To clear the MPLS stripping counters and label entries, perform these tasks:

Command	Purpose
clear mpls strip label dynamic	Clears dynamic label entries from the MPLS label table.
clear counters mpls strip	Clears all MPLS stripping counters.

The following example shows how to clear all MPLS stripping counters:

```
switch# clear counters mpls strip
switch# show mpls strip labels
MPLS Strip Labels:
    Total      : 15000
    Static     : 2
Legend:     * - Static Label
    Interface - where label was first learned
    Idle-Age  - Seconds since last use
    SW-Counter- Packets received in Software
    HW-Counter- Packets switched in Hardware
--------------------------------------------------------------------------------
     Label    Interface      Idle-Age   SW-Counter       HW-Counter
--------------------------------------------------------------------------------
     4096    Eth1/44             15          0                0
     8192    Eth1/44             17          0                0
    12288    Eth1/44             15          0                0
    16384    Eth1/44             39          0                0
    20480    Eth1/44             47          0                0
    24576    Eth1/44              7          0                0
    28672    Eth1/44              5          0                0
    36864    Eth1/44              7          0                0
    40960    Eth1/44             19          0                0
    45056    Eth1/44              9          0                0
    49152    Eth1/44             45          0                0
    53248    Eth1/44              9          0                0
```

Configuration Examples for MPLS Stripping

This example shows how to add static MPLS labels:

```
switch# configure terminal
switch(config)# mpls strip label 100
```

```
switch(config)# mpls strip label 200
switch(config)# mpls strip label 300
```

Additional References

Related Documents

Related Topic	Document Title
IP ACLs	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
MAC ACLs	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
Port-channel symmetric hashing	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
Remote monitoring (RMON)	Configuring RMON, on page 181
Switched port analyzer (SPAN)	Configuring SPAN, on page 225
Troubleshooting	*Cisco Nexus 9000 Series NX-OS Troubleshooting Guide*

CHAPTER 22

Configuring MPLS Access Lists

Configuring MPLS Access Lists

MPLS Access lists enables filtering of MPLS packets based on MPLS label and sending filtered packets to configured redirect interfaces.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**]**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)# feature-set mpls`
	`switch(config)# feature mpls segment-routing`
		Enables parsing of MPLS packets. This is mandatory to filter MPLS packets based on MPLS label.

Step 3	**mpls access list mpls-acl**
	Example:
	`switch(config)# mpls access list mpls-acl`
	`switch(config-mpls-acl)# 10 permit mpls 1600 any redirect Ethernet1/15`
		Configures mpls-access list with filtering based on incoming outer MPLS label.
		In this example, MPLS packets with incoming label 1600 matched and are redirected to Ethernet1/15.

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

Verifying the MPLS Access Lists Configuration

To display the MPLS access list configuration, perform the following task:

Command	Purpose
show mpls access lists	Displays information about MPLS access lists.

Configuration Examples for MPLS Access Lists

This example shows how to configure MPLS access lists:

```
switch# configure terminal
switch(config)# install feature-set mpls
switch(config)# feature-set mpls
switch(config)# feature mpls segment-routing
switch(config)# mpls access list mpls-acl
switch(config-mpls-acl)# 10 permit mpls 1600 any redirect Ethernet1/15
switch(config)# copy running-config startup-config
```

CHAPTER 23

Configuring Graceful Insertion and Removal

This chapter describes how to configure graceful insertion and removal (GIR) on the Cisco Nexus 9000 Series switches.

This chapter contains the following sections:

About Graceful Insertion and Removal

You can use graceful insertion and removal to gracefully eject a switch and isolate it from the network in order to perform debugging or upgrade operations. The switch is removed from the regular forwarding path with minimal traffic disruption. When you are finished performing debugging or upgrade operations, you can use graceful insertion to return the switch to its fully operational (normal) mode.

When you place the switch in maintenance mode, all configured Layer 3 control-plane protocols are isolated from the network. Directly connected routes are not withdrawn or modified during this state. When normal mode is restored, the advertisement of all routes is restored.

In graceful removal, all protocols and vPC domains are gracefully brought down and the switch is isolated from the network. In graceful insertion, all protocols and vPC domains are restored.

The following protocols are supported (for both IPv4 and IPv6 address families):

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Intermediate System-to-Intermediate System (ISIS)

- Open Shortest Path First (OSPF)
- Protocol Independent Multicast (PIM)
- Routing Information Protocol (RIP)

Note For graceful insertion and removal, the PIM protocol is applicable only to vPC environments. During graceful removal, the vPC forwarding role is transferred to the vPC peer for all northbound sources of multicast traffic.

Profiles

By default, the system isolates all enabled protocols during graceful removal and restores them during graceful insertion. The protocols are isolated and restored in a predefined order.

If you want to isolate, shut down, or restore the protocols individually (or perform additional configurations), you can create a profile with configuration commands that can be applied during graceful removal or graceful insertion. However, you need to make sure that the order of the protocols is correct and any dependencies are considered.

The switch supports the following profiles:

- Maintenance-mode profile—Contains all the commands that will be executed during graceful removal, when the switch enters maintenance mode.
- Normal-mode profile—Contains all the commands that will be executed during graceful insertion, when the switch returns to normal mode.

The following commands (along with any configuration commands) are supported in the profiles.

Note The **shutdown** command takes precedence when both **shutdown** and **isolate** are configured under a routing protocol instance or maintenance-mode profile.

Command	Description
isolate	Isolates the protocol from the switch and puts the protocol in maintenance mode.
no isolate	Restores the protocol and puts the protocol in normal mode.
shutdown	Shuts down the protocol or vPC domain.
no shutdown	Brings up the protocol or vPC domain.
system interface shutdown [**exclude fex-fabric**]	Shuts down the system interfaces (except the management interface).
no system interface shutdown [**exclude fex-fabric**]	Brings up the system interfaces.

Command	Description
sleep instance *instance-number seconds*	Delays the execution of the command by a specified number of seconds. You can delay multiple instances of the command. The range for the *instance-number* and *seconds* arguments is from 0 to 2177483647.
python instance *instance-number uri* [*python-arguments*] Example: **python instance 1 bootflash://script1.py**	Configures Python script invocations to the profile. You can add multiple invocations of the command to the profile. You can enter a maximum of 32 alphanumeric characters for the Python arguments.

Snapshots

In Cisco NX-OS, a snapshot is the process of capturing the running states of selected features and storing them on persistent storage media.

Snapshots are useful to compare the state of a switch before graceful removal and after graceful insertion. The snapshot process consists of three parts:

- Creating a snapshot of the states of a few preselected features on the switch and storing them on the persistent storage media
- Listing the snapshots taken at various time intervals and managing them
- Comparing snapshots and showing the differences between features

Guidelines and Limitations for GIR

Graceful Insertion and Replacement have the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.2(1), L2 Graceful Insertion and Replacement is supported. When moving from normal to maintenance mode, MCT goes down resulting in north to south traffic convergence. Zero packet loss is not supported. The following table provides an example of traffic convergence of 10 vPCs with 2 port member on each VPC port and 60k mac scale.

  Table 25:

Trigger	Role	North to South Traffic	South to North Traffic
Normal to maintenance mode	Primary	760 ms	1320 ms

Maintenance mode to normal	Primary	13155 ms	27980 ms
Normal to maintenance mode	Secondary	300 ms	1375 ms
Maintenance mode to normal	Secondary	15905 ms	23350 ms

- Beginning with Cisco NX-OS Release 9.2(1), if you configure the isolate option for OSPF, direct routes and stub routes are advertised as max-metric routes. As a result, north-to-south traffic to the SVI hosts goes through the vPC peer when only one vPC switch is isolated.
- Remove all existing custom profiles before creating new custom profiles for normal-mode and maintenance-mode.

GIR Workflow

Follow these steps to complete the graceful insertion and removal (GIR) workflow:

1. (Optional) Create the maintenance-mode profile. (See Configuring the Maintenance-Mode Profile, on page 332.)
2. (Optional) Create the normal-mode profile. (See Configuring the Normal-Mode Profile, on page 334.)
3. Take a snapshot before triggering graceful removal. (See Creating a Snapshot, on page 335.)
4. Trigger graceful removal to put the switch in maintenance mode. (See Triggering Graceful Removal, on page 339.)
5. Trigger graceful insertion to return the switch to normal mode. (See Triggering Graceful Insertion, on page 341.)
6. Take a snapshot after triggering graceful insertion. (See Creating a Snapshot, on page 335.)
7. Use the **show snapshots compare** command to compare the operational data before and after the graceful removal and insertion of the switch to make sure that everything is running as expected. (See Verifying the GIR Configuration, on page 343.)

Configuring the Maintenance-Mode Profile

You can create a maintenance-mode profile with configuration commands that can be applied during graceful removal or graceful insertion.

Note During the maintenance mode the SVI becomes UP after the reload. In this scenario, use the **isolate include-local** command under router BGP or keep interfaces in shutdown state through maintenance mode to avoid the impact of advertising connected/static routes.

Procedure

	Command or Action	Purpose

Step 1	[**no**] **configure maintenance profile maintenance-mode**
	
	Example:
	`switch# configure maintenance profile maintenance-mode`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config-mm-profile)#`
		Enters a configuration session for the maintenance-mode profile. The **no** option deletes the maintenance profile maintenance-mode.
		
		Depending on which protocols you have configured, you must now enter the appropriate commands to bring down the protocols. For a list of supported commands, see Profiles, on page 330.

Step 2	**end**
	
	Example:
	`switch(config-mm-profile)# end`
	`switch#`
		Closes the maintenance-mode profile.

Step 3	**show maintenance profile maintenance-mode** **Example:** `switch# show maintenance profile maintenance-mode`	Displays the details of the maintenance-mode profile.

Example

This example shows how to create a maintenance-mode profile:

```
switch# configure maintenance profile maintenance-mode
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-mm-profile)# ip pim isolate
switch(config-mm-profile)# router bgp 100
switch(config-mm-profile-router)# shutdown
switch(config-mm-profile)# router eigrp 10
switch(config-mm-profile-router)# shutdown
switch(config-mm-profile-router)# address-family ipv6 unicast
switch(config-mm-profile-router-af)# shutdown
switch(config-mm-profile)# vpc domain 10
switch(config-mm-profile-config-vpc-domain)# shutdown
switch(config-mm-profile)# system interface shutdown
switch(config-mm-profile)# end
Exit maintenance profile mode.
switch# show maintenance profile maintenance-mode
[Maintenance Mode]
ip pim isolate
router bgp 100
  shutdown
router eigrp 10
  shutdown
  address-family ipv6 unicast
    shutdown
vpc domain 10
  shutdown
system interface shutdown
```

This example shows how to configure sleep instance in a custom profile to add a delay before the next protocol change.

```
switch# configure maintenance profile maintenance-mode
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-mm-profile)# router bgp 65001
switch(config-mm-profile-router)# isolate
switch(config-mm-profile-router)# sleep instance 1 10
switch(config-mm-profile)# router eigrp 200
switch(config-mm-profile-router)# isolate
switch(config-mm-profile-router)# sleep instance 2 15
switch(config-mm-profile)# router ospf 100
switch(config-mm-profile-router)# isolate
switch(config-mm-profile-router)# sleep instance 3 20
switch(config-mm-profile)# router ospfv3 300
switch(config-mm-profile-router)# isolate
switch(config-mm-profile-router)# sleep instance 4 5
switch(config-mm-profile)# router isis 400
switch(config-mm-profile-router)# isolate
switch(config-mm-profile)#end
Exit maintenance profile mode.
switch#
```

Note If you need to run exec commands or add a dynamic delay while the maintenance mode profile is applied, use the **python instance** *instance-number uri* [*python-arguments*] script.

Configuring the Normal-Mode Profile

You can create a normal-mode profile with configuration commands that can be applied during graceful removal or graceful insertion.

Procedure

	Command or Action	Purpose

Step 1	[**no**] **configure maintenance profile normal-mode**
	Example:
	`switch# configure maintenance profile normal-mode`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config-mm-profile)#`
		Enters a configuration session for the normal-mode profile. The **no** version removes the maintenance profile normal-mode.
		Depending on which protocols you have configured, you must now enter the appropriate commands to bring up the protocols. For a list of supported commands, see Profiles, on page 330.

Step 2	**end**
	Example:
	`switch(config-mm-profile)# end`
	`switch#`
		Closes the normal-mode profile.

Step 3	**show maintenance profile normal-mode**	Displays the details of the normal-mode profile.

	Command or Action	Purpose
	Example: `switch# show maintenance profile normal-mode`	

Example

This example shows how to create a maintenance profile normal-mode:

```
switch# configure maintenance profile normal-mode
switch(config-mm-profile)# no system interface shutdown
switch(config-mm-profile)# vpc domain 10
switch(config-mm-profile-config-vpc-domain)# no shutdown
switch(config-mm-profile)# router eigrp 10
switch(config-mm-profile-router)# no shutdown
switch(config-mm-profile-router)# address-family ipv6 unicast
switch(config-mm-profile-router-af)# no shutdown
switch(config-mm-profile)# router bgp 100
switch(config-mm-profile-router)# no shutdown
switch(config-mm-profile)# no ip pim isolate
switch(config-mm-profile)# end
Exit maintenance profile mode.
switch# show maintenance profile normal-mode
[Normal Mode]
no system interface shutdown
vpc domain 10
  no shutdown
  router eigrp 10
    no shutdown
address-family ipv6 unicast
  no shutdown
router bgp 100
  no shutdown
no ip pim isolate
```

Creating a Snapshot

You can create a snapshot of the running states of selected features. When you create a snapshot, a predefined set of **show** commands are run and the outputs are saved.

Procedure

	Command or Action	Purpose

Step 1	**snapshot create** *snapshot-name description*
	Example:
	`switch# snapshot create snap_before_maintenance Taken before maintenance`
	`Executing 'show interface'... Done`
	`Executing 'show ip route summary vrf all'... Done`
	`Executing 'show ipv6 route summary vrf`
		Captures the running state or operational data of selected features and stores the data on persistent storage media.
		You can enter a maximum of 64 alphanumeric chapters for the snapshot name and a maximum of 254 alphanumeric characters for the description.

	Command or Action	Purpose

	`all'... Done`
	`Executing 'show bgp sessions vrf all'... Done`
	`Executing 'show ip eigrp topology summary'... Done`
	`Executing 'show ipv6 eigrp topology summary'... Done`
	`Feature 'vpc' not enabled, skipping...`
	`Executing 'show ip ospf vrf all'... Done`
	`Feature 'ospfv3' not enabled, skipping...`
	`Feature 'isis' not enabled, skipping...`
	`Feature 'rip' not enabled, skipping...`
	`Snapshot 'snap_before_maintenance' created`
		Use the **snapshot delete** {**all** | *snapshot-name*} command to delete all snapshots or a specific snapshot.

Step 2	**show snapshots**
	Example:
	`switch# show snapshots`
	`Snapshot Name            Time                Description`
	`-----------------------------------------------------------`
	`snap_before_maintenance   Wed Aug 19 13:53:28 2015  Taken before maintenance`
		Displays snapshots present on the switch.

Step 3	**show snapshots compare** *snapshot-name-1 snapshot-name-2* [**summary** | **ipv4routes** | **ipv6routes**] **Example:** `switch# show snapshots compare snap_before_maintenance snap_after_maintenance`	Displays a comparison of two snapshots. The **summary** option displays just enough information to see the overall changes between the two snapshots. The **ipv4routes** and **ipv6routes** options display the changes in IPv4 and IPv6 routes between the two snapshots.

Example

The following example shows a summary of the changes between two snapshots:

```
switch# show snapshots compare snapshot1 snapshot2 summary
feature                             snapshot1      snapshot2     changed
basic summary
    # of interfaces                 16             12            *
    # of vlans                      10             4             *
    # of ipv4 routes                33             3             *
    …………

interfaces
    # of eth interfaces             3              0             *
    # of eth interfaces up          2              0             *
    # of eth interfaces down        1              0             *
    # of eth interfaces other       0              0

    # of vlan interfaces            3              1             *
    # of vlan interfaces up         3              1             *
    # of vlan interfaces down       0              0
    # of vlan interfaces other      0              1             *
    …………
```

The following example shows the changes in IPv4 routes between two snapshots:

```
switch# show snapshots compare snapshot1 snapshot2 ipv4routes
metric                          snapshot1      snapshot2       changed
# of routes                     33             3               *
# of adjacencies                10             4               *

Prefix             Changed Attribute
------             -----------------
23.0.0.0/8         not in snapshot2
10.10.10.1/32      not in snapshot2
21.1.2.3/8         adjacency index has changed from 29 (snapshot1) to 38 (snapshot2)
………

There were 28 attribute changes detected
```

Adding Show Commands to Snapshots

You can specify additional **show** commands to be captured in snapshots. These **show** commands are defined in user-specified snapshot sections.

Procedure

	Command or Action	Purpose
Step 1	**snapshot section add** *section "show-command" row-id element-key1* [*element-key2*] **Example:** `switch# snapshot section add myshow "show ip interface brief" ROW_intf intf-name`	Adds a user-specified section to snapshots. The *section* is used to name the **show** command output. You can use any word to name the section. The **show** command must be enclosed in quotation marks. Non-**show** commands will not be accepted. The *row-id* argument specifies the tag of each row entry of the **show** command's XML output. The *element-key1* and *element-key2* arguments specify the tags used to distinguish among row entries. In most cases, only the *element-key1* argument needs to specified to be able to distinguish among row entries. **Note** To delete a user-specified section from snapshots, use the **snapshot section delete** *section* command.
Step 2	**show snapshots sections** **Example:** `switch# show snapshots sections`	Displays the user-specified snapshot sections.
Step 3	**show snapshots compare** *snapshot-name-1 snapshot-name-2* [**summary** | **ipv4routes** | **ipv6routes**]	Displays a comparison of two snapshots.

	Command or Action	Purpose
	Example: `switch# show snapshots compare snap1 snap2`	The **summary** option displays just enough information to see the overall changes between the two snapshots. The **ipv4routes** and **ipv6routes** options display the changes in IPv4 and IPv6 routes between the two snapshots.

Example

The following example adds the **show ip interface brief** command to the myshow snapshot section. It also compares two snapshots (snap1 and snap2) and shows the user-specified sections in both snapshots.

```
switch# snapshot section add myshow "show ip interface brief" ROW_intf intf-name
switch# show snapshots sections
user-specified snapshot sections
--------------------------------
[myshow]
  cmd:  show ip interface brief
  row:  ROW_intf
  key1: intf-name
  key2: -

[sect2]
  cmd:  show ip ospf vrf all
  row:  ROW_ctx
  key1: instance_number
  key2: cname

switch# show snapshots compare snap1 snap2
=============================================================================
Feature              Tag                  snap1                snap2
=============================================================================

[bgp]
-----------------------------------------------------------------------------
............

[interface]
-----------------------------------------------------------------------------

       [interface:mgmt0]
                     vdc_lvl_in_pkts      692310            **692317**
                     vdc_lvl_in_mcast     575281            **575287**
                     vdc_lvl_in_bcast     77209             **77210**
                     vdc_lvl_in_bytes     63293252          **63293714**
                     vdc_lvl_out_pkts     41197             **41198**
                     vdc_lvl_out_ucast    33966             **33967**
                     vdc_lvl_out_bytes    6419714           **6419788**
............

[ospf]
-----------------------------------------------------------------------------
............

[myshow]
-----------------------------------------------------------------------------
```

```
       [interface:Ethernet1/1]
              state                 up                   **down**
              admin_state           up                   **down**
............
```

Triggering Graceful Removal

In order to perform debugging or upgrade operations, you can trigger a graceful removal of the switch, which will eject the switch and isolate it from the network.

Before you begin

If you want the system to use a maintenance-mode profile that you create, see Configuring the Maintenance-Mode Profile, on page 332.

Procedure

	Command or Action	Purpose

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

Step 2	**system mode maintenance** [**dont-generate-profile** | **timeout** *value* | **shutdown** | **on-reload reset-reason** *reason*]
	Example:
	`switch(config)# system mode maintenance`
	`Following configuration will be applied:`
	
	`   ip pim isolate`
	`   router bgp 65502`
	`     isolate`
	`   router ospf p1`
	`     isolate`
	`   router ospfv3 p1`
	`     isolate`
	
	`Do you want to continue (y/n)? [no] y`
	
	`Generating a snapshot before going into`
	` maintenance mode`
	
	`Starting to apply commands...`
	
	`Applying : ip pim isolate`
	`Applying : router bgp 65502`
	`Applying :   isolate`
	`Applying : router ospf p1`
	`Applying :   isolate`
	`Applying : router ospfv3 p1`
		Puts all enabled protocols in maintenance mode (using the **isolate** command).
		The following options are available:
		• **dont-generate-profile**—Prevents the dynamic searching of enabled protocols and executes commands configured in a maintenance-mode profile. Use this option if you want the system to use a maintenance-mode profile that you have created.
		• **timeout** *value*—Keeps the switch in maintenance mode for a specified number of minutes. The range is from 5 to 65535. Once the configured time elapses, the switch returns to normal mode automatically. The **no system mode maintenance timeout** command disables the timer.
		• **shutdown**—Shuts down all protocols, vPC domains, and interfaces except the management interface (using the **shutdown** command). This option is

	Command or Action	Purpose

	`Applying :   isolate`
	
	`Maintenance mode operation successful.`
		disruptive while the default (which uses the **isolate** command) is not.
		
		• **on-reload reset-reason** *reason*—Boots the switch into maintenance mode automatically in the event of a specified system crash. The **no system mode maintenance on-reload reset-reason** command prevents the switch from being brought up in maintenance mode in the event of a system crash.
		
		The maintenance mode reset reasons are as follows:
		• HW_ERROR—Hardware error
		• SVC_FAILURE—Critical service failure
		• KERN_FAILURE—Kernel panic
		• WDOG_TIMEOUT—Watchdog timeout
		• FATAL_ERROR—Fatal error
		• LC_FAILURE—Line card failure
		• MATCH_ANY—Any of the above reasons
		
		The system prompts you to continue. Enter **y** to continue or **n** to terminate the process.

Step 3	(Optional) **show system mode**
	
	Example:
	`switch(config)# show system mode`
	`System Mode: Maintenance`
		Displays the current system mode.
		
		The switch is in maintenance mode. You can now perform any desired debugging or upgrade operations on the switch.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration. This command is required if you want to preserve maintenance mode following a reboot.

Example

This example shows how to shut down all protocols, vPC domains, and interfaces on the switch:

```
switch(config)# system mode maintenance shutdown

Following configuration will be applied:
```

```
   vpc domain 10
     shutdown
   router bgp 65502
     shutdown
   router ospf p1
     shutdown
   router ospfv3 p1
     shutdown
   system interface shutdown

Do you want to continue (y/n)? [no] y

Generating a snapshot before going into maintenance mode

Starting to apply commands...

Applying : vpc domain 10
Applying :   shutdown
Applying : router bgp 65502
Applying :   shutdown
Applying : router ospf p1
Applying :   shutdown
Applying : router ospfv3 p1
Applying :   shutdown

Maintenance mode operation successful.
```

This example shows how to automatically boot the switch into maintenance mode if a fatal error occurs:

```
switch(config)# system mode maintenance on-reload reset-reason fatal_error
```

Triggering Graceful Insertion

When you finish performing any debugging or upgrade operations, you can trigger a graceful insertion to restore all protocols.

Before you begin

If you want the system to use a normal-mode profile that you create, see Configuring the Maintenance-Mode Profile, on page 332.

Procedure

	Command or Action	Purpose

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

Step 2	**no system mode maintenance** [**dont-generate-profile**]	Puts all enabled protocols in normal mode (using the **no isolate** command).

	Command or Action	Purpose

	Example:
	`switch(config)# no system mode maintenance dont-generate-profile`
	`Following configuration will be applied:`
	
	`no ip pim isolate`
	`router bgp 65502`
	`no isolate`
	`router ospf p1`
	`no isolate`
	`router ospfv3 p1`
	`no isolate`
	
	`Do you want to continue (y/n)? [no]` **y**
	
	`Starting to apply commands...`
	
	`Applying : no ip pim isolate`
	`Applying : router bgp 65502`
	`Applying :   no isolate`
	`Applying : router ospf p1`
	`Applying :   no isolate`
	`Applying : router ospfv3 p1`
	`Applying :   no isolate`
	
	`Maintenance mode operation successful.`
	
	`Generating Current Snapshot`
		The **dont-generate-profile** option prevents the dynamic searching of enabled protocols and executes commands configured in a normal-mode profile. Use this option if you want the system to use a normal-mode profile that you have created.
		
		The system prompts you to continue. Enter **y** to continue or **n** to terminate the process.

Step 3	(Optional) **show system mode**
	
	Example:
	`switch(config)# show system mode`
	`System Mode: Normal`
		Displays the current system mode. The switch is now in normal mode and is fully operational.

Maintenance Mode Enhancements

Starting with Release 7.0(3)I5(1), the following maintenance mode enhancements have been added to Cisco Nexus 9000 Series switches:

- In the system maintenance shutdown mode, the following message is added:

  ```
  NOTE: The command system interface shutdown will shutdown all interfaces excluding mgmt
   0.
  ```

- Entering the CLI command, **system mode maintenance** checks and sends alerts for the orphan ports.
- In isolate mode, when the vPC is configured, the following message is added:

  ```
  NOTE: If you have vPC orphan interfaces, please ensure vpc orphan-port suspend is
  configured under them, before proceeding further.
  ```

- Custom Profile Configuration: A new CLI command, **system mode maintenance always-use-custom-profile** is added for custom profile configuration. A new CLI command, **system mode maintenance non-interactive** is added for Cisco Nexus 9000 Series switches only. It provides a way to facilitate the transition to maintenance mode or normal mode without confirmation being done or each step being printed on the CLI session.

When a loopback interface is configured with an IP address on a device, and this device is advertised to a peer device, then the device (with the loopback interface) moves to maintenance mode. In such a case, use the custom maintenance profile when **system interface shutdown** is configured on the device.

When you create a custom profile (in maintenance or normal mode), it displays the following message:

```
Please use the command system mode maintenance always-use-custom-profile
if you want to always use the custom profile.
```

- A delay has been added before the after_maintenance snapshot is taken. The **no system mode maintenance** command exits once all the configuration for the normal mode has been applied, the mode has been changed to normal mode, and a timer has been started to take the after_maintenance snapshot. Once the timer expires, the after_maintenance snapshot is taken in the background and a new warning syslog, MODE_SNAPSHOT_DONE is sent once the snapshot is complete.

  The final output of the CLI command **no system mode maintenance** indicates when the after_maintenance snapshot is generated:

  ```
  The after_maintenance snapshot will be generated in <delay> seconds.
  After that time, please use show snapshots compare before_maintenance
  after_maintenance to check the health of the system. The timer delay
  for the after_maintenance snapshot is defaulted to 120 seconds but it
  can be changed by a new configuration command.
  ```

  The new configuration command to change the timer delay for the after_maintenance snapshot is **system mode maintenance snapshot-delay <seconds>**. This configuration overrides the default setting of 120 seconds to any value between 0 and 65535 and it is displayed in the ASCII configuration.

  A new show command, **show maintenance snapshot-delay** has also been added to display the current snapshot-delay value. This new show command supports the XML output.

- A visible CLI indicator has been added to display when the system is in the maintenance mode, for example, `switch(maint-mode)#`.

- Support for the SNMP traps has been added when the device moves from the maintenance mode to the normal mode and vice-versa through CLI reload, or system reset. The **snmp-server enable traps mmode cseMaintModeChangeNotify** trap is added to enable changing to the maintenance mode trap notification. The **snmp-server enable traps mmode cseNormalModeChangeNotify** is added to enable changing to the normal mode trap notification. Both the traps are disabled by default.

Verifying the GIR Configuration

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

Command	Purpose
show interface brief	Displays abbreviated interface information.
show maintenance on-reload reset-reasons	Displays the reset reasons for which the switch comes up in maintenance mode. For a description of the maintenance mode reset reasons, see Triggering Graceful Removal, on page 339.

Command	Purpose
show maintenance profile [**maintenance-mode** | **normal-mode**]	Displays the details of the maintenance-mode or normal-mode profile.
show maintenance timeout	Displays the maintenance-mode timeout period, after which the switch automatically returns to normal mode.
show {**running-config** | **startup-config**} **mmode** [**all**]	Displays the maintenance-mode section of the running or startup configuration. The **all** option includes the default values.
show snapshots	Displays snapshots present on the switch.
show snapshots compare *snapshot-name-1 snapshot-name-2* [**summary** | **ipv4routes** | **ipv6routes**]	Displays a comparison of two snapshots. The **summary** option displays just enough information to see the overall changes between the two snapshots. The **ipv4routes** and **ipv6routes** options display the changes in IPv4 and IPv6 routes between the two snapshots.
show snapshots dump *snapshot-name*	Displays the content of each file that was generated when the snapshot was taken.
show snapshots sections	Displays the user-specified snapshot sections.
show system mode	Displays the current system mode.

Configuration Examples for GIR

The **redistribute direct** configuration under Border Gateway Protocol (BGP) will attract traffic as the BGP isolate mode does not withdraw direct routes. This example shows how to use the **route-map** command to enable BGP to withdraw direct routes in isolate mode.

Policy Configuration

Use the **route-map my-rmap-deny** command in maintenance mode to exclude SVIs with a tag 200 configuration.

```
switch(config)# route-map my-rmap-deny deny 10
switch(config-route-map)# match tag 200
switch(config-route-map)# exit
switch(config)# route-map my-rmap-deny permit 20
```

Use the **route-map my-rmap-permit** command in normal mode to include SVIs with a tag 200 configuration.

```
switch(config)# route-map my-rmap-permit permit 10
switch(config-route-map)# match tag 200
switch(config-route-map)# exit
switch(config)# route-map my-rmap-permit permit 20
```

Virtual IP (vIP)/Switch Virtual Interface (SVI) Configuration

```
switch(config)# interface loopback 200
switch(config-if)# ip address 192.0.2.100/8 tag 200
switch(config)# interface vlan 2
switch(config-if)# ip address 192.0.2.108/8 tag 200
....
switch(config)# interface vlan 3
switch(config-if)# ip address 192.0.2.102/8 tag 200
```

BGP Configuration

```
switch(config)# feature bgp
switch(config)# router bgp 100
switch(config-router)# neighbor 192.0.2.100
....
```

Maintenance Mode Profile

```
switch# configure maintenance profile maintenance-mode
switch(config-mm-profile)# router bgp 200
switch(config-mm-profile-router)# address-family ipv4 unicast
switch(config-mm-profile-router-af)# redistribute direct route-map my-rmap-deny
switch(config-mm-profile-router-af)# exit
switch(config-mm-profile)# sleep instance 1 10
```

Normal Mode Profile

```
switch# configure maintenance profile normal-mode
switch(config-mm-profile)# router bgp 100
switch(config-mm-profile-router)# address-family ipv4 unicast
switch(config-mm-profile-router-af)# redistribute direct route-map my-rmap-permit
switch(config-mm-profile-router-af)# exit
switch(config-mm-profile)# sleep instance 1 20
```

CHAPTER 24

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.

For information on Cisco NX-OS optionality feature, 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.

> **Note** Beginning with Cisco NX-OS Release 7.0(3)I2(1), SMU package files have an .rpm extension. Earlier files have a .bin extension.

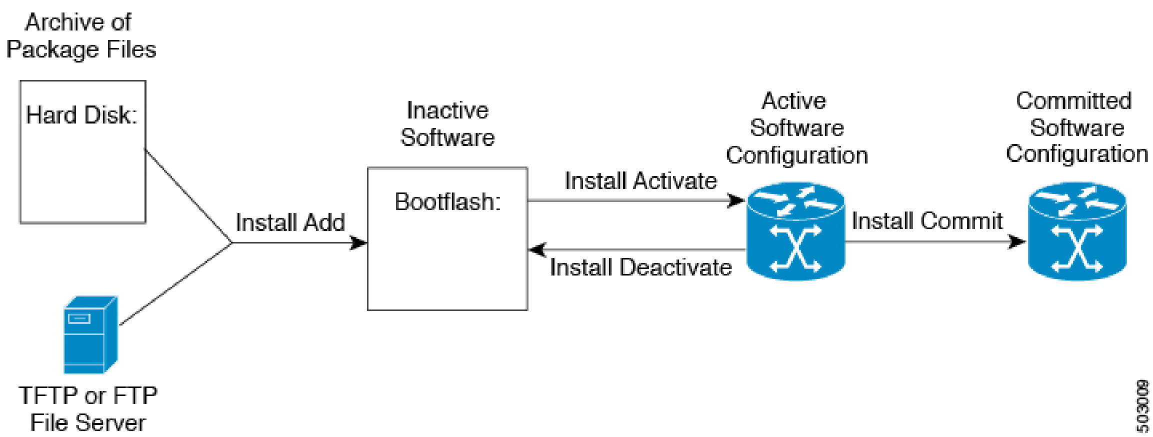
Package Management

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 6: 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.
- SMU packages being activated must be compatible with the image version running in the switch.
- Activation is performed only after the package compatibility checks have been passed. If a conflict is found, an error message is displayed.
- 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.

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.

	Command or Action	Purpose
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.
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 26: 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/ nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm 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.

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>copy ftp://username:password@hostname-or-ipaddress/directory-path/filename bootflash:

<code>switch# copy ftp://john:secret@10.1.1.1/images/
nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
 bootflash:</code></td><td>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.</td></tr>
</table>

Command	Purpose
copy sftp://*hostname-or-ipaddress***/***directory-path***/***filename* **bootflash:** `switch# copy sftp://10.1.1.1/images/nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm 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 This procedure uses Cisco NX-OS CLI commands to add and activate RPM package files. If you would prefer to use YUM commands, follow the instructions in the "Installing RPMs from Bash" section of the Cisco Nexus 9000 Series NX-OS Programmability Guide.

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 352.

Procedure

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session to the console port.
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: nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm`	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. You can use the **install add** command with a tarball SMU to install multiple SMUs at the same time.

	Command or Action	Purpose
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.

Step 5	Required: **install activate** *filename*
	Example:
	`switch# install activate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm`
	Example:
	`switch# install activate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm`
	`Install operation 18 !!WARNING!! This patch`
	`will get activated only after a reload of`
	`the switch.`
	`at Wed Jun 22 00:42:12 2016`
		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.

Example

This example shows how to add multiple SMU package files with a tarball and then verify the added package files.

```
switch# install add bootflash:nxos.CSC123456-n9k_ALL-1.0.0-7.0.3.I7.3.
lib32_n9000.tar
[####################] 100%
Install operation 882 completed successfully at Tue Mar  6 17:30:31 2018

switch#
switch# show install inactive
Boot Image:
        NXOS Image: bootflash:////nxos.7.0.3.I7.3.bin-219-CCO

Inactive Packages:
        nxos.CSC123456_core-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000
        nxos.CSC123456_eth-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000


Inactive Base Packages:

switch#
```

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 355.

Procedure

	Command or Action	Purpose
Step 1	**install commit** *filename* **Example:** `switch# install commit nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm`	Commits the current set of packages so that these packages are used if the device is restarted.
Step 2	(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 nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 2 completed successfully at Wed Jun 22 01:20:46 2016

switch# show install committed
Committed Packages:
nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
```

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.

Note This procedure uses Cisco NX-OS CLI commands to deactivate and remove RPM package files. If you would prefer to use YUM commands, follow the instructions in the "Erasing an RPM" section of the Cisco Nexus 9000 Series NX-OS Programmability Guide.

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.
Step 2	**install deactivate** *filename* **Example:** `switch# install deactivate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm`	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 3	(Optional) **show install inactive** **Example:** `switch# show install inactive`	Displays the inactive packages on the device.
Step 4	(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 5	(Optional) **install remove** {*filename* | **inactive**} **Example:** `switch# install remove nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm Proceed with removing`	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.

	Command or Action	Purpose

	`nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm? (y/n)? [n] y`
	Example:
	`switch# install remove inactive`
	`Proceed with removing? (y/n)? [n] y`
		• 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 remove** command with the **inactive** keyword.

Example

This example shows how to deactivate a package, commit the changes, and remove the inactive package from the device:

```
switch# install deactivate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 3 completed successfully at Wed Jun 22 01:20:36 2016

switch# show install inactive
Inactive Packages:
nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm

switch# install commit
Install operation 4 completed successfully at Wed Jun 22 01:20:46 2016

switch# install remove nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Proceed with removing nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm? (y/n)? [n]
 y
Install operation 5 completed successfully at Wed Jun 22 01:20:57 2016
```

This example shows how to deactivate multiple packages with one command, remove the inactive packages from the device, and verify the package removal:

```
switch# install deactivate nxos.CSC123456_core-n9k_ALL-1.0.0-7.0.3.I7.3.
lib32_n9000 nxos.CSC123456_eth-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000
[####################] 100%
Install operation 884 completed successfully at Tue Mar  6 17:34:02 2018

switch#
switch# show install inactive
Boot Image:
        NXOS Image: bootflash:////nxos.7.0.3.I7.3.bin-219-CCO

Inactive Packages:
        nxos.CSC123456_core-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000
        nxos.CSC123456_eth-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000


Inactive Base Packages:

switch#
switch# install remove nxos.CSC123456_core-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000
Proceed with removing nxos.CSC123456_core-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000? (y/n)?  [n]
 y
[####################] 100%
```

```
Install operation 885 completed successfully at Tue Mar  6 17:34:56 2018

switch# install remove nxos.CSC123456_eth-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000
Proceed with removing nxos.CSC123456_eth-n9k_ALL-1.0.0-7.0.3.I7.3.lib32_n9000? (y/n)?  [n]
 y
[####################] 100%
Install operation 886 completed successfully at Tue Mar  6 17:35:14 2018

switch#
switch# show install inactive
Boot Image:
        NXOS Image: bootflash:///nxos.7.0.3.I7.3.bin-219-CCO

Inactive Packages:


Inactive Base Packages:

switch#
```

Downgrading Feature RPMs

Follow this procedure to downgrade an installed feature RPM to the base feature RPM.

Note This procedure uses Cisco NX-OS CLI commands to downgrade feature RPMs. If you would prefer to use YUM commands, follow the instructions in the "Downgrading an RPM" section of the Cisco Nexus 9000 Series NX-OS Programmability Guide.

Procedure

	Command or Action	Purpose

Step 1	(Optional) **show install packages**
	Example:
	`switch# show install packages`
	`ntp.lib32_n9000     1.0.1-7.0.3.I2.2e`
	`    installed`
		Displays the feature RPM packages on the device.

Step 2	Required: **run bash**
	Example:
	`switch# run bash`
	`bash-4.2$`
		Loads Bash.

Step 3	Required: **cd /rpms** **Example:** `bash-4.2$ cd /rpms`	Changes to the RPMs folder in Bash.

Step 4	Required: **ls ****feature********
	Example:
	`bash-4.2$ ls *ntp*`
	`ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm`
		Lists the RPM for the specified feature.

	Command or Action	Purpose
Step 5	Required: **cp** *filename* **/bootflash** **Example:** `bash-4.2$ cp ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm /bootflash`	Copies the base feature RPM to the bootflash.
Step 6	Required: **exit** **Example:** `bash-4.2$ exit`	Exits Bash.

Step 7	Required: **install add bootflash:***filename* **activate downgrade**
	
	Example:
	`switch# install add bootflash:ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm activate downgrade`
	`Adding the patch (/ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm)`
	`[#############       ]  60%`
	`Adding the patch (/ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm)`
	`[####################] 100%`
	`Install operation 11 completed successfully at Thu Sep  8 15:35:35 2015`
	
	`Activating the patch (/ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm)`
	`This install operation requires system reload. Do you wish to continue (y/n)?: [n] y`
	`[  217.975959] [1473348971]  writing reset reason 132, System reset due to reload patch(es) activation`
	`[  217.991166] [1473348971]\ufffd\ufffd`
	`CISCO SWITCH Ver7.51`
	`Device detected on 0:6:0 after 0 msecs`
	`Device detected on 0:1:1 after 0 msecs`
	`Device detected on 0:1:0 after 0 msecs`
	`MCFrequency 1333Mhz`
	`Relocated to memory`
		Downgrades the feature RPM.
		
		Note If you are prompted to reload the device, enter **Y**. A reload is required only when downgrading the NTP and SNMP feature RPMs.

Step 8	(Optional) **show install packages | i** *feature*
	
	Example:
	`switch# show install packages | i ntp`
	`ntp.lib32_n9000    1.0.0-7.0.3.I2.2e installed`
		Displays the base feature RPM on the device.

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
Wed Jun 22 01:26:09 2016
Install operation 1 by user 'admin' at Wed Jun 22 01:19:19 2016
Install add bootflash:nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 1 completed successfully at Wed Jun 22 01:19:24 2016
----------------------------------------
Install operation 2 by user 'admin' at Wed Jun 22 01:19:29 2016
Install activate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 2 completed successfully at Wed Jun 22 01:19:45 2016
----------------------------------------
Install operation 3 by user 'admin' at Wed Jun 22 01:20:05 2016
Install commit nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 3 completed successfully at Wed Jun 22 01:20:08 2016
----------------------------------------
Install operation 4 by user 'admin' at Wed Jun 22 01:20:21 2016
Install deactivate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 4 completed successfully at Wed Jun 22 01:20:36 2016
----------------------------------------
Install operation 5 by user 'admin' at Wed Jun 22 01:20:43 2016
Install commit nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 5 completed successfully at Wed Jun 22 01:20:46 2016
----------------------------------------
Install operation 6 by user 'admin' at Wed Jun 22 01:20:55 2016
Install remove nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 6 completed successfully at Wed Jun 22 01:20:57 2016
----------------------------------------
Install operation 7 by user 'admin' at Wed Jun 22 01:21:07 2016
Install remove
Install operation 7 completed successfully at Wed Jun 22 01:21:10 2016
```

This example shows how to display additional information, including any impact to nodes and processes:

```
switch# show install log detail
Wed Jun 22 01:24:03 2016
Install operation 1 by user 'admin' at Wed Jun 22 01:19:19 2016
Installer started downloading the package:
/nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
via bootflash
Install add bootflash:nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Copying file at Wed Jun 22 01:19:20 2016
Download success, 238545 bytes received
Verifying package
Checking MD5 at Wed Jun 22 01:19:21 2016
MD5 checksum OK
Checking HW platform at Wed Jun 22 01:19:22 2016
Checking SW platform at Wed Jun 22 01:19:23 2016
Package verified successfully
Sending patch file to plugin manager at Wed Jun 22 01:19:23 2016
The following package is now available to be activated: nxos.CSCab00001-n9k_ALL-
1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 1 completed successfully at Wed Jun 22 01:19:24 2016
----------------------------------------
```

```
Install operation 2 by user 'admin' at Wed Jun 22 01:19:29 2016
Install activate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
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 Wed Jun 22 01:19:45 2016
----------------------------------------
Install operation 3 by user 'admin' at Wed Jun 22 01:20:05 2016
Install commit nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
MD5 checksum OK for patch: nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 3 completed successfully at Wed Jun 22 01:20:08 2016
----------------------------------------
Install operation 4 by user 'admin' at Wed Jun 22 01:20:21 2016
Install deactivate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
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 Wed Jun 22 01:20:36 2016
----------------------------------------
Install operation 5 by user 'admin' at Wed Jun 22 01:20:43 2016
Install commit nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
MD5 checksum OK for patch: nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 5 completed successfully at Wed Jun 22 01:20:46 2016
----------------------------------------
Install operation 6 by user 'admin' at Wed Jun 22 01:20:55 2016
Install remove nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
Install operation 6 completed successfully at Wed Jun 22 01:20:57 2016
----------------------------------------
Install operation 7 by user 'admin' at Wed Jun 22 01:21:07 2016
Install remove
Install operation 7 completed successfully at Wed Jun 22 01:21:10 2016
```

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 Wed Jun 22 00:42:10 2016
Install activate nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm
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 Wed Jun 22 00:42:12 2016
```

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 351.
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 352.

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) [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`
		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.
		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.

	Command or Action	Purpose

	` 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+'`	

Additional References

Related Documents

Related Topic	Document Title
Software upgrades	*Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*

CHAPTER 25

Performing Configuration Replace

This chapter includes the following sections:

About Configuration Replace and Commit-timeout

The configuration replace feature enables you to replace the running configuration of the Cisco Nexus switch with the user provided configuration without reloading the device. The device reload may be required only when a configuration itself requires a reload. The running configuration file that is provided by the user should be taken using copy running file. Unlike **copy file: to running**, the configuration replace feature is not a merge operation. This feature replaces the entire running configuration with a new configuration that is provided by the user. If there is a failure in the configuration replace, the original configuration is restored in the switch. From Cisco NX-OS Release 9.3(1), **best-effort** option is introduced. This option enables the configuration replace to execute the full patch despite any error in the commands and the original configuration is not restored in the switch.

The commit-timeout feature enables you to rollback to the previous configuration after successfully performing the configuration replace operation. If the commit timer expires, the rollback operation is automatically initiated.

Note

- You must provide a valid running configuration that has been received with the Cisco NX-OS device. It should not be a partial configuration.

Overview

The configuration replace feature has the following operation steps:

- Configuration replace intelligently calculates the difference between the current running-configuration and the user-provided configuration in the Cisco Nexus switch and generates a patch file which is the difference between the two files. You can view this patch file which includes a set of configuration commands.
- Configuration replace applies the configuration commands from the patch file similarly to executing commands.
- The configuration rolls back to or restores the previous running configuration under the following situations:
    - If there is a mismatch in the configuration after the patch file has been applied.
    - If you perform the configuration operation with a commit timeout and the commit timer expires.
- The configuration does not roll back to or does not restore the previous running configuration when the best-effort option is used. This option enables the configuration replace to execute the full patch despite any error in the commands and will not roll back to the previous configuration.
- You can view the exact configuration that caused a failure using the **show config-replace log exec** command.
- Restore operations that fail while restoring the switch to the original configuration, are not interrupted. The restore operation continues with the remaining configuration. Use the **show config-replace log exec** command to list the commands that failed during the restore operation.
- If you enter the **configure replace commit** command before the timer expires, the commit timer stops and the switch runs on the user provided configuration that has been applied through the configuration replace feature.
- If the commit timer expires, roll back to the previous configuration is initiated automatically.
- In Cisco NX-OS Release 9.3(1), semantic validation support is added for the configuration replace. This semantic validation is done as part of the precheck in configuration replace. The patch gets applied only when the semantic validation is successful. After applying the patch file, configuration replace triggers the verification process. The configuration replace compares the running-configuration with the user configuration file during the verification process. If there is a mismatch, it restores the device to the original configuration.

The differences between configuration replace and copying a file to the running-configuration are as follows:

Configuration Replace	Copying a file
The **configure replace** *<target-url>* command removes the commands from the current running-configuration that are not present in the replacement file. It also adds commands that need to be added to the current running-configuration.	The **copy** *<source-url>* **running-config** command is a merge operation which preserves all the commands from, both the source file and the current running-configuration. This command does not remove the commands from the current running-configuration that are not present in the source file.
You must use a complete Cisco NX-OS configuration file as the replacement file for the **configure replace** *<target-url>* command.	You can use a partial configuration file as a source file for the **copy** *<source-url>* **running-config** command.

Benefits of Configuration Replace

The benefits of configuration replace are:

- You can replace the current running-configuration file with the user-provided configuration file without having to reload the switch or manually undo CLI changes to the running-configuration file. As a result, the system downtime is reduced.
- You can revert to the saved Cisco NX-OS configuration state.
- It simplifies the configuration changes by allowing you to apply a complete configuration file to the device, where only the commands that need to be added or removed are affected. The other service and configurations that are not modified remain untouched.
- If you configure the commit-timeout feature, you can rollback to the previous configuration even when the configuration replace operation has been successful.

Guidelines and Limitations for Configuration Replace

The configuration replace feature has the following configuration guidelines and limitations:

- The configuration replace feature is supported on Cisco Nexus 3000 Series and Cisco Nexus 9000 Series switches.
- Only one user can perform the configuration replace, checkpoint, and rollback operations, or copy the running-configuration to the startup configuration at the same time. Parallel operations such as operations via multiple Telnet, SSH, or NX-API sessions are not supported. The multiple configuration replace or rollback request is serialized, for example, only after the first request is completed, processing of the second request begins.
- You are not allowed to initiate another configuration replace operation when the commit timer is running. You must either stop the timer by using the **configure replace commit** command or wait until the commit timer expires before you initiate another configuration replace operation.
- When **system default switchport shutdown** or **no system default switchport shutdown** is used with **configure replace bootflash:target_config_file** command, the user should make sure that desired port state (shutdown or no shutdown) statement is present in the target_config_file for all switchport interfaces.
- The commit-timeout feature is initiated only if you perform the configuration replace operation with the commit-timeout. The timer value range is from 30 to 3600 seconds.
- The user provided configuration file must be the valid show running-configuration output that is taken from the Cisco NX-OS device (copy run file). The configuration cannot be a partial configuration and must include mandated commands, such as user admin and so on.
- We do not recommend a configuration replace operation that is performed on the configuration file that is generated across the software version because this operation could fail. A new configuration file must be regenerated whenever there is change in the software version.
- The configuration replace operation is not supported if you attempt to replace a multichassis EtherChannel trunk (MCT) configuration with a virtual peer-link configuration. This operation is not allowed because the physical MCT uses the CFS distribution over Ethernet mode and the virtual peer-link use the CFS distribution over IP mode.

- We recommend that you do not change any configuration from others sessions if the configuration replace operation is in progress because it could cause the operation to fail.
- Note the following about the configuration replace feature:
    - The configuration replace feature is not supported on Cisco Nexus 9500 platform switches with -R line cards.
    - The configuration replace feature is not supported on the hardware profile port mode feature on Cisco Nexus 92160YC-X and Cisco Nexus 93180LC-EX switches.
    - The configuration replace feature is supported **only** for the configure terminal mode commands. The configure profile, configure jobs, and any other modes are not supported.
    - The configuration replace feature could fail if the running configuration includes the **feature-set mpls** or the **mpls static range** commands and tries to move to a configuration without MPLS or modifies the label range.
    - The configuration replace feature does not support autoconfigurations.
- If the line card to which the configuration replace feature is applied is offline, the configuration replace operation fails.
- An ITD service must be shut down (**shutdown**) prior to making ITD changes with the configuration replace feature.
- Entering maintenance mode from the user configuration is not supported.
- Using the **configure replace** command from maintenance mode asks for a user-confirmation with the following warning:

```
Warning: System is in maintenance mode. Please ensure user config won't inadvertently
revert back config in maintenance mode profile.
            Do you wish to proceed anyway? (y/n)  [n]
```

- Using the **configure replace** command from maintenance mode with a *<non-interactive>* option is supported. It takes the *yes* user-confirmation by default and proceeds.
- If your configurations demand reloading the Cisco NX-OS device in order to apply the configuration, then you must reload these configurations after the configuration replace operation.
- The order of the commands in the user provided configuration file must be the same as those commands in the running configuration of the Cisco Nexus switch.
- The user configuration file to which you need to replace the running configuration on the switch using CR should be generated from the running-config of the switch after configuring the new commands. The user configuration file should not be manually edited with the CLI commands and the sequence of the configuration commands should not be altered.
- The semantic validation is not supported in 4-Gig memory platforms.
- When different versions of a feature are present in the running configuration and user configuration (for example: VRRPv2 and VRRPv3), semantic validation option does not work as expected. This issue is a known limitation.

Recommended Workflow for Configuration Replace

The following workflow is the recommended workflow for configuration replace:

Performing a Configuration Replace

To perform configuration replace, do the following:

Procedure

	Command or Action	Purpose
Step 1	**configure replace** { < *uri_local* > | < *uri_remote* > } [**verbose** | **show-patch**]	Performs configuration replace. If you make the configuration changes through any sessions when configuration replace is in progress, the configuration replace operation fails. If you send a configuration replace request when one configuration request is already in progress, then it gets serialized.
Step 2	**configure replace** [**bootflash** / **scp** / **sftp**] < *user-configuration-file* > **show-patch**	Displays the differences between the running-configuration and the user-provided configuration.
Step 3	**configure replace** [**bootflash** / **scp** / **sftp**] < *user-configuration-file* > **verbose**	Replaces the configuration on the switch with the new user configuration that is provided by the user. Configuration replace is always atomic.
Step 4	**configure replace** <*user-configuration-file*> [**best-effort**]	Replaces the configuration on the switch with the new user configuration and enables the configuration replace with semantic validation. The best-effort option enables the configuration replace to execute the full patch despite any error in the commands and also make sure that the previous configuration is not rolled back.
Step 5	**configure replace** <*user-configuration-file*> [**verify-and-commit**]	Replaces the configuration on the switch with the new user configuration and enables the configuration replace with semantic validation. The verify-and-commit option is used for enabling the semantic validation. Patch will be executed only if semantic validation of the full patch gets passed. You can use the best-effort option or the verify-and-commit option or both the options at the same time.

	Command or Action	Purpose
Step 6	**configure replace** *<user-configuration-file>* [**verify-only**]	Shows only the patch and does Semantic validation on the patch, and display the results. The patch does not get applied to the system.
Step 7	(Optional) **configure replace** [**bootflash** / **scp** / **sftp**] < *user-configuration-file* > **verbose commit-timeout** <*time*>	Configures the commit time in seconds. The timer starts after the configuration replace operation is successfully completed.
Step 8	(Optional) **configure replace** [**commit**]	Stops the commit timer and continues the configuration replace configuration. **Note** This step is applicable only if you have configured the commit-timeout feature. **Note** To rollback to the previous configuration, you must wait for the expiry of the commit timer. Once the timer expires, the switch is automatically rolled back to the previous configuration.
Step 9	(Optional) **configure replace** [**bootflash**/**scp**/**sftp**] *<user-configuration-file> non-interactive*	There is no user prompt in maintenance mode. The **yes** user-confirmation is taken by default, and rollback proceeds. You can use the non-interactive option only in the maintenance mode.

Verifying Configuration Replace

To check and verify configuration replace and its status, use the commands that are outlined in the table:

Table 27: Verifying Configuration Replace

Command	Purpose
configure replace [**bootflash**/**scp**/**sftp**]<*user-configuration-file*] **show-patch**	Displays the difference between the running-configurations and user-provided configurations.
show config-replace log exec	Displays a log of all the configurations executed and those that failed. In case of an error, it displays an error message against that configuration.
show config-replace log verify	Displays the configurations that failed, along with an error message. It does not display configurations that were successful.

Command	Purpose
show config-replace status	Displays the status of the configuration replace operations, including in-progress, successful, and failure. If you have configured the commit-timeout feature, the commit and timer status and the commit timeout time remaining is also displayed.

Examples for Configuration Replace

See the following configuration examples for configuration replace:

- Use the **configure replace bootflash:** *<file>* **show-patch** CLI command to display the difference between the running-configurations and user-provided configurations.

```
switch(config)# configure replace bootflash:<file> show-patch
Collecting Running-Config
Converting to checkpoint file
#Generating Rollback Patch
!!
no role name abc
```

- Use the **configure replace bootflash:** *<file>* **verbose** CLI command to replace the entire running-configuration in the switch with the user-configuration.

```
switch(config)# configure replace bootflash:<file> verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
========================================================
config t
no role name abc
========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

Sample Example with adding of BGP configurations.
 switch(config)# sh run | section bgp
 switch(config)# sh file bootflash:file | section bgp
feature bgp
router bgp 1
    address-family ipv4 unicast
    neighbor 1.1.1.1
switch(config)#
switch(config)# configure replace bootflash:file verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
========================================================
```

```
config t
feature bgp
router bgp 1
address-family ipv4 unicast
neighbor 1.1.1.1
=========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

switch(config)# sh run | section bgp
feature bgp
router bgp 1
  address-family ipv4 unicast
  neighbor 1.1.1.1

Sample Example with ACL
 switch(config)# configure replace bootflash:run_1.txt
 Collecting Running-Config
 Generating Rollback patch for switch profile
 Rollback Patch is Empty
 Note: Applying config parallelly may fail Rollback verification
 Collecting Running-Config
 #Generating Rollback Patch
 Executing Rollback Patch
 =========================================================
 config t
 no ip access-list nexus-50-new-xyz
 ip access-list nexus-50-new-xyz-jkl-abc
 10 remark Newark
 20 permit ip 17.31.5.0/28 any
 30 permit ip 17.34.146.193/32 any
 40 permit ip 17.128.199.0/27 any
 50 permit ip 17.150.128.0/22 any
 =========================================================
 Generating Running-config for verification
 Generating Patch for verification

 Rollback completed successfully.

 switch(config)#


 switch(config)# show run aclmgr | sec nexus-50-new-xyz-jkl-abc
 ip access-list nexus-50-new-xyz-jkl-abc
   10 remark Newark
   20 permit ip 17.31.5.0/28 any
   30 permit ip 17.34.146.193/32 any
   40 permit ip 17.128.199.0/27 any
   50 permit ip 17.150.128.0/22 any
```

- Use the **configure replace bootflash:user-config.cfg verify-only** CLI command to generate and verify the patch semantically.

```
switch(config)# configure replace bootflash:user-config.cfg verify-only

Version match between user file and running configuration.
Pre-check for User config PASSED
Collecting Running-Config
Converting to checkpoint file
Generating Rollback Patch
Validating Patch
=========================================================
```

```
`config t `
`interface Ethernet1/1`
`shutdown`
`no switchport trunk allowed vlan`
`no switchport mode`
`no switchport`
`exit`
Skip non dme command for CR validation
`interface Vlan1`
`shutdown`
`interface Ethernet1/1`
`shutdown`
`no switchport`
`ip address 1.1.1.1/24`
`exit`
Skip non dme command for CR validation
=============================================
Patch validation completed successful
switch(config)#
```

- Use the **configure replace bootflash:user-config.cfg best-effort verify-and-commit** CLI command to replace the switch running configuration with the given user configuration after performing the sematic validation on patch.

```
switch(config)# configure replace bootflash:user-config.cfg best-effort verify-and-commit


 Version match between user file and running configuration.
Pre-check for User config PASSED
ADVISORY: Config Replace operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Config Replace failure.

Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Collecting Running-Config
Generating Rollback Patch

Validating Patch
Patch validation completed successful
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Rollback Patch

Configure replace completed successfully. Please run 'show config-replace log exec' to
 see if there is any configuration that requires reload to take effect.

switch(config)#
```

- Use the **show config-replace log exec** CLI command to check all the configuration that is executed and failures if any.

```
switch(config)# show config-replace log exec
Operation            : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme               : tmp
Rollback done By     : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
--------------------------------------------------------------------------------
```

```
time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
End Time             : Wed, 06:39:47 25 Jan 2017
Rollback Status      : Success

Executing Patch:
----------------
switch#config t
switch#no role name abc
```

- Use the **show config-replace log verify** CLI command to check the failed configuration if any.

```
switch(config)# show config-replace log verify
Operation            : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme               : tmp
Rollback done By     : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
End Time             : Wed, 06:39:47 25 Jan 2017
Status               : Success

Verification patch contains the following commands:
---------------------------------------------------
!!
! No changes
--------------------------------------------------------------------------------

time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
```

- Use the **show config-replace status** CLI command to check the status of configuration replace.

```
switch(config)# show config-replace status
Last operation : Rollback to file
Details:
  Rollback type: atomic replace_tmp_28081
  Start Time: Wed Jan 25 06:39:28 2017
  End Time: Wed Jan 25 06:39:47 2017
  Operation Status: Success
switch(config)#
```

CHAPTER 26

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 that are 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 cause 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*

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 2579	*Textual Conventions for SMIv2*
RFC 2819	*Remote Network Monitoring Management Information Ba*
RFC 2863	*The Interfaces Group MIB*
RFC 3176	*InMon Corporation's sFlow: A Method for Monitoring Traf and Routed Networks*
RFC 3411 and RFC 3418	*An Architecture for Describing Simple Network Managem (SNMP) Management Frameworks*
RFC 3413	*Simple Network Management Protocol (SNMP) Applicati*
RFC 3417	*Transport Mappings for the Simple Network Managemen (SNMP)*

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

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
```

```
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
```

Embedded Event Manager System Events and Configuration Examples

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*.

INDEX