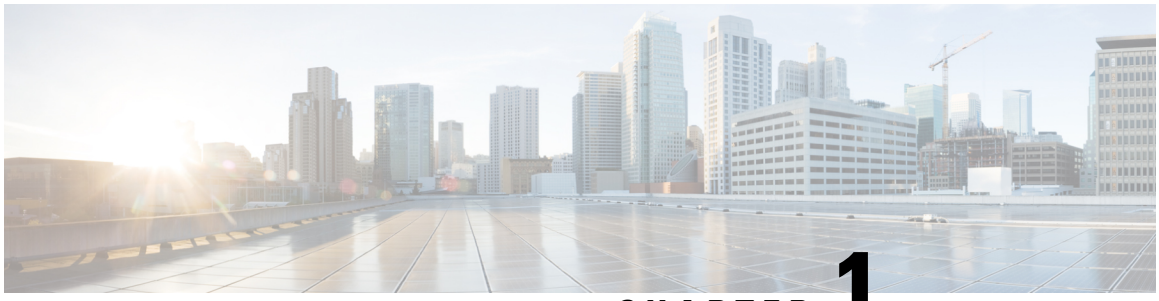
Cisco Nexus 3400-S NX-OS System Management Configuration Guide 9.3(x)

First Published: 2019-12-23

Last Modified: 2020-08-26

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



CHAPTER 1

New and Changed Information

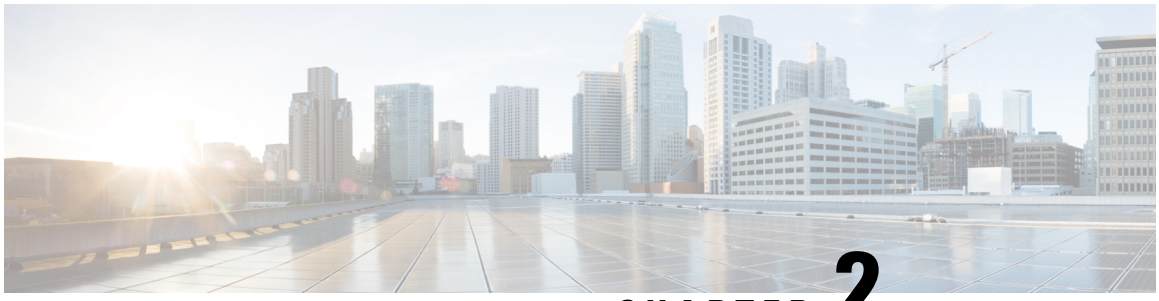
This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3400-S Series NX-OS System Management Configuration Guide, Release 9.3(x)*.

New and Changed Information

Table 1: New and Changed Features for Cisco NX-OS Release 9.3(x)

Feature	Description	Changed in Release	Where Documented
Event Log Auto-Collection and Backup	Updates to the auto-collection YAML file and additional options for the bloggerd log-snapshot command.	9.3(5)	Event Log Auto-Collection and Backup, on page 118
LLDP Multi-Neighbor and Port-Channel Interface Support	Support for up to three (3) LLDP neighbors per interface. Support for LLDP on interface port channels.	9.3(5)	LLDP Multi-Neighbor Support, on page 174 Enabling or Disabling LLDP Support on Port-Channel Interfaces, on page 176
PortLoopback Diagnostic	Support for the PortLoopback test for Runtime or Health Monitoring and On-Demand Diagnostics tests.	9.3(5)	Runtime or Health Monitoring Diagnostics, on page 88
Modified Repeated System Logging Message Format	Support for an updated indicator in repeated syslog messages.	9.3(5)	Repeated System Logging Messages, on page 48
Extended Event Log Storage	Introduced support for extended on-switch and off-switch event logging file storage.	9.3(3)	Event Log Auto-Collection and Backup, on page 118
System Management	First release	9.3(3)	

New and Changed Information



CHAPTER 2

Overview

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

This chapter includes the following sections:

Software Image

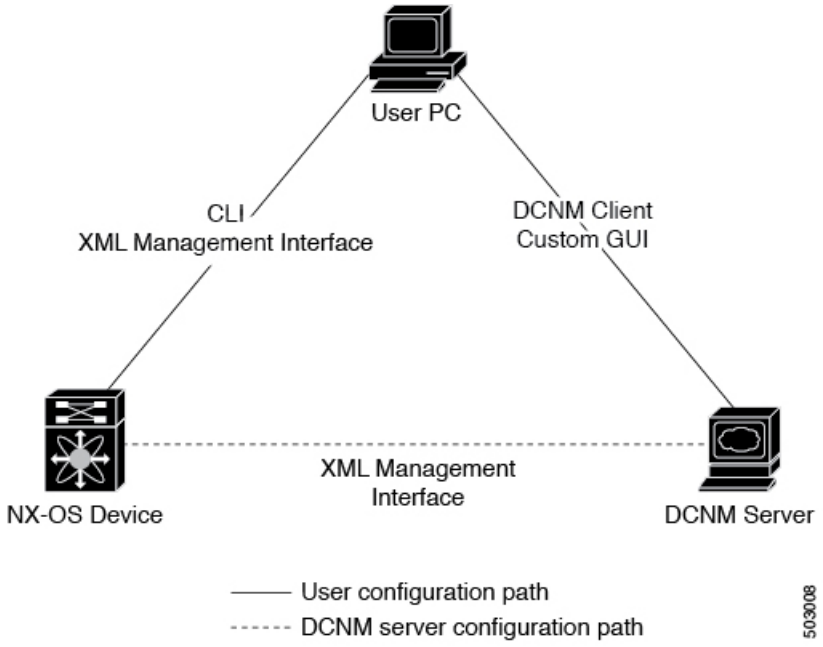
The Cisco NX-OS software consists of one NXOS software image. This image runs on all Cisco Nexus 3400 Series switches.

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-S 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 3400 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

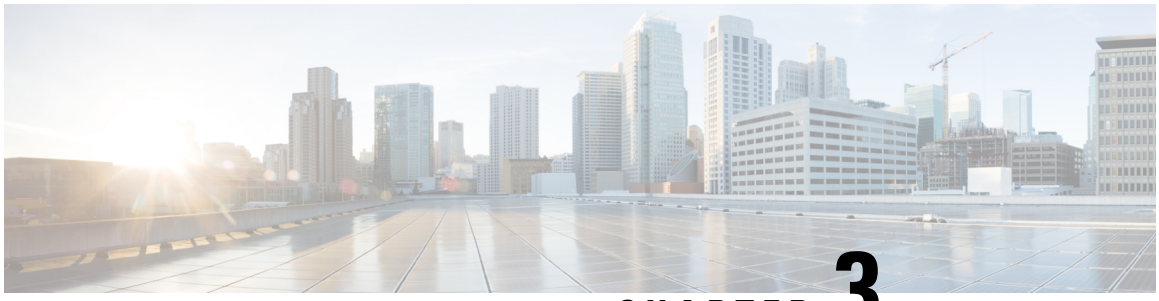
Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 3400 Series switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.

Troubleshooting Features

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

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

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



CHAPTER 3

Configuring NTP

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

This chapter includes the following sections:

About NTP

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

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

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

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

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

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

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

NTP Associations

An NTP association can be one of the following:

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

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

NTP as a Time Server

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

Clock Manager

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

The clock manager allows you to specify the protocol to control the various clocks in the system. Once you specify the protocol, the system clock starts updating. For information on configuring the clock manager, see the Cisco Nexus 3400 Series NX-OS Fundamentals Configuration Guide.

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 3400 Series NX-OS Unicast Routing Configuration Guide for more information about VRFs.

Licensing Requirements for NTP

Product	License Requirement
Cisco NX-OS	NTP requires no license. Any feature not included in a license package is bundled with the nx-os i provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing sche Cisco NX-OS Licensing Guide.

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.
- 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature ntp**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

Configuring the Device as an Authoritative NTP Server

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

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp master** [*stratum*]
3. (Optional) **show running-config ntp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp server** {*ip-address* | *ipv6-address* | *dns-name*} [**key** *key-id*] [**maxpoll** *max-poll*] [**minpoll** *min-poll*] [**prefer**] [**use-vrf** *vrf-name*]
3. [**no**] **ntp peer** {*ip-address* | *ipv6-address* | *dns-name*} [**key** *key-id*] [**maxpoll** *max-poll*] [**minpoll** *min-poll*] [**prefer**] [**use-vrf** *vrf-name*]
4. (Optional) **show ntp peers**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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).

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

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp authentication-key** *number* **md5** *md5-string*
3. **ntp server** *ip-address* **key** *key-id*
4. (Optional) **show ntp authentication-keys**
5. [**no**] **ntp trusted-key** *number*
6. (Optional) **show ntp trusted-keys**
7. [**no**] **ntp authenticate**
8. (Optional) **show ntp authentication-status**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

	Command or Action	Purpose
Step 5	[**no**] **ntp trusted-key** *number* **Example:** `switch(config)# ntp trusted-key 42`	Specifies one or more keys (defined in Step 2) that an unconfigured remote symmetric, broadcast, and multicast time source must provide in its NTP packets in order for the device to synchronize to it. The range for trusted keys is from 1 to 65535. This command provides protection against accidentally synchronizing the device to a time source that is not trusted.
Step 6	(Optional) **show ntp trusted-keys** **Example:** `switch(config)# show ntp trusted-keys`	Displays the configured NTP trusted keys.
Step 7	[**no**] **ntp authenticate** **Example:** `switch(config)# ntp authenticate`	Enables or disables authentication for ntp passive, ntp broadcast client, and ntp multicast. NTP authentication is disabled by default.
Step 8	(Optional) **show ntp authentication-status** **Example:** `switch(config)# show ntp authentication-status`	Displays the status of NTP authentication.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring NTP Access Restrictions

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

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

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 descending order:

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

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp access-group match-all** | {{**peer** | **serve** | **serve-only** | **query-only** }*access-list-name*}
3. (Optional) **show ntp access-groups**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

	Command or Action	Purpose
		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. • 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.
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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp source** *ip-address*
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp source-interface** *interface*
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

Configuring NTP Logging

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

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ntp logging**
3. (Optional) **show ntp logging-status**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 logging** **Example:** `switch(config)# ntp logging`	Enables or disables system logs to be generated with significant NTP events. NTP logging is disabled by default.
Step 3	(Optional) **show ntp logging-status** **Example:** `switch(config)# show ntp logging-status`	Displays the NTP logging configuration status.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the NTP Configuration

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

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

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

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

Configuration Examples for NTP

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

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

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

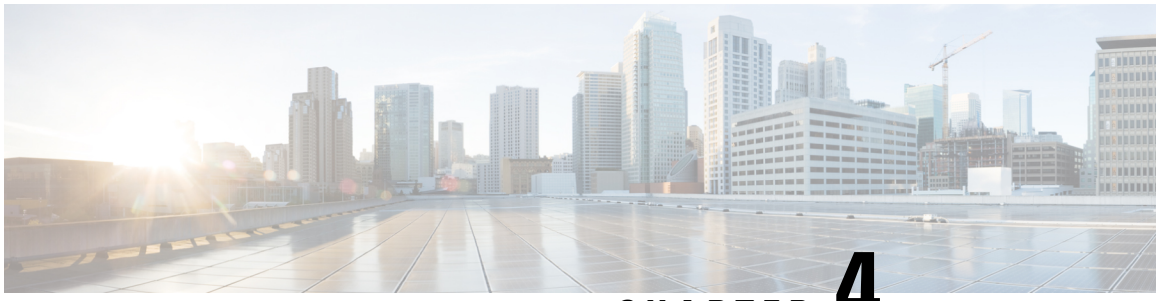
- Peer restrictions are applied to IP addresses that pass the criteria of the access list named “peer-acl.”
- Serve restrictions are applied to IP addresses that pass the criteria of the access list named “serve-acl.”
- Serve-only restrictions are applied to IP addresses that pass the criteria of the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

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

Additional References

Related Documents

Related Topic	Document Title
Clock manager	Cisco Nexus 3400 Series NX-OS Fundamentals Configuration Guide



CHAPTER 4

Configuring CDP

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

This chapter includes the following sections:

About CDP

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

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

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

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

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

- Device ID
- Address
- Port ID
- Capabilities

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

All CDP packets include a VLAN ID. If you configure CDP on a Layer 2 access port, the CDP packets sent from that access port include the access port VLAN ID. If you configure CDP on a Layer 2 trunk port, the CDP packets sent from that trunk port include the lowest configured VLAN ID allowed on that trunk port. The trunk port can receive CDP packets that include any VLAN ID in the allowed VLAN list for that trunk port. For more information on VLANs, see the Cisco Nexus 3400 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.

Virtualization Support

Cisco NX-OS supports one instance of CDP.

Licensing Requirements for CDP

Product	License Requirement
Cisco NX-OS	CDP requires no license. Any feature not included in a license package is bundled with the nx-os imag provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for CDP

CDP has the following configuration guidelines and limitations:

- CDP can discover up to 256 neighbors per port if the port is connected to a hub with 256 connections.
- CDP must be enabled on the device or you cannot enable it on any interfaces.
- You can configure CDP on physical interfaces and port channels only.

Default Settings for CDP

This table lists the default settings for CDP parameters.

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

Configuring CDP

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

Enabling or Disabling CDP Globally

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

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

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **cdp enable**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	[**no**] **cdp enable** **Example:** `switch(config)# cdp enable`	Enables or disables the CDP feature on the entire device. It is enabled by default.
Step 3	(Optional) **copy running-config startup-config** **Example:** `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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot*/*port*
3. [**no**] **cdp enable**
4. (Optional) **show cdp interface** *interface slot*/*port*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **cdp advertise** {**v1** | **v2**}
3. (Optional) **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}
4. (Optional) **cdp holdtime** *seconds*
5. (Optional) **cdp timer** *seconds*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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**.

	Command or Action	Purpose
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.
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.

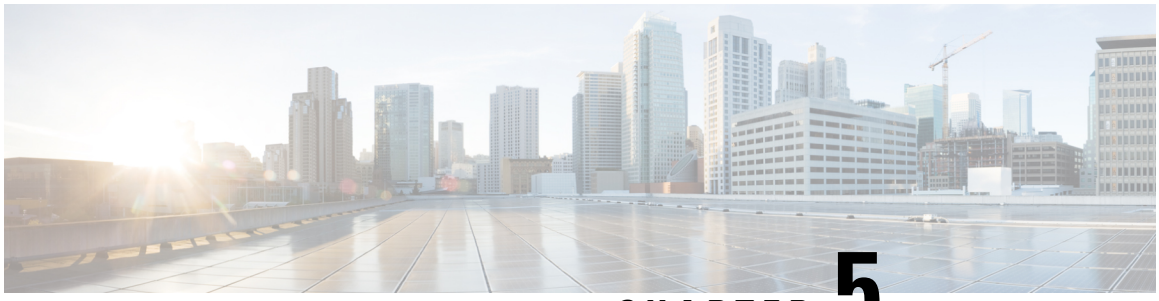
Use the **clear cdp counters** command to clear CDP statistics on an interface.

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

Configuration Example for CDP

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

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



CHAPTER 5

Configuring System Message Logging

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

This chapter contains the following sections:

About System Message Logging

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

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

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

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

Table 3: System Message Severity Levels

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

Level	Description
4 – warning	Warning condition
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

The syslog server can be configured 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.

Licensing Requirements for System Message Logging

Product	License Requirement
Cisco NX-OS	System message logging requires no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the Cisco NX-OS Licensing Guide.

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.
- Syslog server can be configured with support for a secure TLS transport connectivity to remote logging servers. This feature supports TLSv1.1 and TLSv1.2.
- 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).

Default Settings for System Message Logging

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

Table 4: Default System Message Logging Parameters

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

Parameters	Default
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.

SUMMARY STEPS

1. **terminal monitor**
2. **configure terminal**
3. [**no**] **logging console** [*severity-level*]
4. (Optional) **show logging console**
5. [**no**] **logging monitor** [*severity-level*]
6. (Optional) **show logging monitor**
7. [**no**] **logging message interface type ethernet description**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

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

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. **logging origin-id** {**hostname** | **ip** *ip-address* | **string** *text-string*}
3. (Optional) **show logging origin-id**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 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 switch-abc`
		Displays the configured hostname, IP address, or text string that is appended to syslog messages that are sent to remote syslog servers.

	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.

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 log:messages.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **logging logfile** *logfile-name severity-level* [**size** *bytes*]
3. **logging event** {**link-status** | **trunk-status**} {**enable** | **default**}
4. (Optional) **show logging info**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 name of the log file used to store system messages and the minimum severity level to log. 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 You can optionally specify a maximum file size.

	Command or Action	Purpose
		The default severity level is 5, and the file size is from 4096 to 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 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **logging module** [*severity-level*]
3. (Optional) **show logging module**
4. [**no**] **logging level** *facility severity-level*
5. (Optional) **show logging level** [*facility*]
6. (Optional) [**no**] **logging level** *ethpm*
7. [**no**] **logging timestamp** {**microseconds** \| **milliseconds** \| **seconds**}
8. (Optional) **show logging timestamp**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

	Command or Action	Purpose
Step 5	(Optional) **show logging level** [*facility*] **Example:** `switch(config)# show logging level aaa`	Displays the logging level configuration and the system default level by facility. If you do not specify a facility, the device displays levels for all facilities.

Step 6	(Optional) [**no**] **logging level** *ethpm*
	Example:
	`switch(config)# logging level ethpm ?`
	`<0-7>`
	`0-emerg;1-alert;2-crit;3-err;4-warn;5-notif;6-inform;7-debug`
	`link-down Configure logging level for link down syslog messages`
	`link-up Configure logging level for link up syslog messages`
	`switch(config)#logging level ethpm link-down ?`
	`error ERRORS`
	`notif NOTICE`
	`(config)# logging level ethpm link-down error ?`
	`<CR>`
	`(config)# logging level ethpm link-down notif ?`
	`<CR>`
	`switch(config)#logging level ethpm link-up ?`
	`error ERRORS`
	`notif NOTICE`
	`(config)# logging level ethpm link-up error ?`
	`<CR>`
	`(config)# logging level ethpm link-up notif ?`
	`<CR>`
		Enables logging of the Ethernet Port Manager link-up/link-down syslog messages at level 3.
		Use the **no** option to use the default logging level for Ethernet Port Manager syslog messages.

Step 7	[**no**] **logging timestamp** {**microseconds** | **milliseconds** | **seconds**} **Example:** `switch(config)# logging timestamp milliseconds`	Sets the logging time-stamp units. By default, the units are seconds. **Note** This command applies to logs that are kept in the switch. It does not apply to the external logging server.
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

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

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 3400 Series NX-OS Unicast Routing Configuration Guide.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **logging server** *host* [*severity-level* [**use-vrf** *vrf-name*]]
3. **logging source-interface loopback** *virtual-interface*
4. (Optional) **show logging server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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::)db*::3 5 use-vrf red`
		Configures a syslog server at the specified hostname or IPv4 or IPv6 address. You can limit logging of messages to a particular VRF by using the **use-vrf** keyword. Severity levels range from 0 to 7:
		• 0 – emergency
		• 1 – alert
		• 2 – critical
		• 3 – error
		• 4 – warning
		• 5 – notification
		• 6 – informational
		• 7 – debugging
		The default outgoing facility is local7.
		The **no** option removes the logging server for the specified host.

	Command or Action	Purpose
		The first example forwards all messages on facility local 7. The second example forwards messages with severity level 5 or lower for 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

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **logging server** *host* [*severity-level* [**port** *port-number*][**secure**[**trustpoint client-identity** *trustpoint-name*]][**use-vrf** *vrf-name*]]
3. (Optional) **logging source-interface** *interface name*
4. (Optional) **show logging server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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* [**port** *port-number*][**secure**[**trustpoint client-identity** *trustpoint-name*]][**use-vrf** *vrf-name*]] **Example:** `switch(config)# logging server 192.0.2.253 secure` **Example:**	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.

	Command or Action	Purpose
	`switch(config)# logging server 2001::3 5 secure trustpoint client-identity myCA use-vrf red`	
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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **crypto ca trustpoint** *trustpoint-name*
3. **crypto ca authenticate** *trustpoint-name*
4. (Optional) **show crypto ca certificate**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

Step 3	Required: **crypto ca authenticate** *trustpoint-name* **Example:** `switch(config-trustpoint)# crypto ca authenticate winca`	Configures a CA certificate for the trustpoint.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. **crypto key generate rsa label** *key name* **exportable modules 2048**
3. [**no**] **crypto ca trustpoint** *trustpoint-name*
4. **rsakeypair** *key-name*
5. **crypto ca trustpoint** *trustpoint-name*
6. [**no**] **crypto ca enroll** *trustpoint-name*
7. **crypto ca import** *trustpoint-name* **certificate**
8. (Optional) **show crypto ca certificates**
9. **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 4	Required: **rsakeypair** *key-name* **Example:** `switch(config-trustpoint)# rsakeypair myKey`	Associates the keypair generated to the trustpoint CA.
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 on a UNIX or Linux System

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

facility.level <five tab characters> *action*

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

Table 5: syslog Fields in syslog.conf

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

Field	Description
Action	Destination for messages, which can be a filename, a hostname preceded by the at sign (@), or a comma-separated list of users or an asterisk (*) for all logged-in users.

SUMMARY STEPS

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:
2. Create the log file by entering these commands at the shell prompt:
3. Make sure that the system message logging daemon reads the new changes by checking myfile.log after entering this command:

DETAILED STEPS

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:

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

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

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

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

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

Displaying and Clearing Log Files

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

SUMMARY STEPS

1. **show logging last** *number-lines*
2. **show logging logfile** [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*]
3. **show logging nvram** [**last** *number-lines*]
4. **clear logging logfile**
5. **clear logging nvram**

DETAILED STEPS

	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** [**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 3	**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 4	**clear logging logfile** **Example:** `switch# clear logging logfile`	Clears the contents of the log file.
Step 5	**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 [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*]	Displays the messages 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.

Command	Purpose
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.

Repeated System Logging Messages

System processes generate logging messages. Depending on the filters used to control which severity levels are generated, a large number of messages can be produced with many of them being repeated.

To make it easier to develop scripts to manage the volume of logging messages, and to eliminate repeated messages from "flooding" the output of the **show logging log** command, the following method of logging repeated messages is used.

In the old method, when the same message was repeated, the default was to state the number of times it reoccurred in the message:

```
2019 Mar 11 13:42:44 Cisco-customer %PTP-2-PTP_INCORRECT_PACKET_ON_SLAVE:
Incorrect delay response packet received on slave interface Eth1/48 by
2c:5a:0f:ff:fe:51:e9:9f. Source Port Identity is 08:00:11:ff:fe:22:3e:4e. Requesting Port
Identity is 00:1c:73:ff:ff:ee:f6:e5
2019 Mar 11 13:43:15 Cisco-customer last message repeated 242 times
```

The new method simply appends the repeat count to the end of the repeated message:

```
2019 Mar 11 13:42:44 Cisco-customer %PTP-2-PTP_INCORRECT_PACKET_ON_SLAVE:
Incorrect delay response packet received on slave interface Eth1/48 by
2c:5a:0f:ff:fe:51:e9:9f. Source Port Identity is 08:00:11:ff:fe:22:3e:4e. Requesting Port
Identity is 00:1c:73:ff:ff:ee:f6:e5

2019 Mar 11 13:43:15 Cisco-customer %PTP-2-PTP_INCORRECT_PACKET_ON_SLAVE:
Incorrect delay response packet received on slave interface Eth1/48 by
2c:5a:0f:ff:fe:51:e9:9f. Source Port Identity is 08:00:11:ff:fe:22:3e:4e. Requesting Port
Identity is 00:1c:73:ff:ff:ee:f6:e5 (message repeated 242 times)
```

Configuration Example for System Message Logging

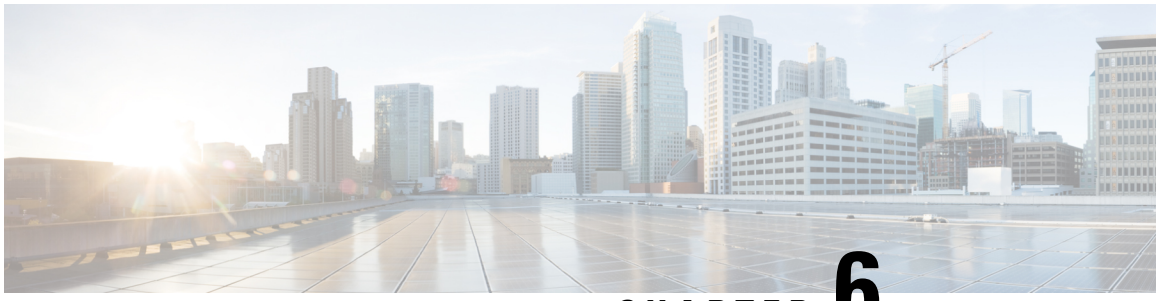
This example shows how to configure system message logging:

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

Additional References

Related Documents

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



CHAPTER 6

Configuring 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

Trap Type	Description
link	: linkUp
rf	: redundancy_framework
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 6: 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 two 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.

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.

Licensing Requirements for SNMP

Product	License Requirement
Cisco NX-OS	SNMP requires no license. Any feature not included in a license package is bundled with the nx-os provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing sche Cisco NX-OS Licensing Guide.

Guidelines and Limitations for SNMP

SNMP has the following configuration guidelines and limitations:

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

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

Default Settings for SNMP

The following table lists the default settings for SNMP parameters.

Parameters	Default
License notifications	Enabled

Configuring SNMP

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

Configuring SNMP Users

You can configure a user for SNMP.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server user** *name* [**auth** {**md5** | **sha**} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*] [**localizedkey**]]
3. (Optional) **show snmp user**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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**} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*] [**localizedkey**]] **Example:** `switch(config)# snmp-server user Admin auth sha abcd1234 priv abcdefgh`	Configures an SNMP user with authentication and privacy parameters. The passphrase can be any case-sensitive, alphanumeric string up to 64 characters. If you use the **localizedkey** keyword, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters. The engineID format is a 12-digit, colon-separated decimal number.
Step 3	(Optional) **show snmp user** **Example:**	Displays information about one or more SNMP users.

	Command or Action	Purpose
	`switch(config) # show snmp user`	
Step 4	(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**.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server user** *name* **enforcePriv**
3. **snmp-server globalEnforcePriv**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server user** *name group*
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

Creating SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server community** *name* {**group** *group* | **ro** | **rw**}
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server community** *name* [**use-ipv4acl** *acl-name* | **use-ipv6acl** *acl-name*]
3. **snmp-server user** *username* [**use-ipv4acl** *acl-name* | **use-ipv6acl** *acl-name*]
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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:**	Assigns an IPv4 or IPv6 ACL to an SNMPv2 community to filter SNMP requests. **Note** IPv6 ACLs are supported for SNMPv2 communities.

	Command or Action	Purpose
	`switch(config)# snmp-server community public use-ipv4acl myacl`	
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.
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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server host** *ip-address* **traps version 1** *community* [**udp_port** *number*]
3. **snmp-server host** *ip-address* {**traps** | **informs**} **version 2c** *community* [**udp_port** *number*]
4. **snmp-server host** *ip-address* {**traps** | **informs**} **version 3** {**auth** | **noauth** | **priv**} *username* [**udp_port** *number*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**snmp-server host** *ip-address* **traps version 1** *community* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 traps version 1 public`	Configures a host receiver for SNMPv1 traps. The *ip-address* can be an IPv4 or IPv6 address. The *community* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.
Step 3	**snmp-server host** *ip-address* {**traps** | **informs**} **version 2c** *community* [**udp_port** *number*] **Example:** `switch(config)# snmp-server host 192.0.2.1 informs version 2c public`	Configures a host receiver for SNMPv2c traps or informs. The *ip-address* can be an IPv4 or IPv6 address. The *community* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server host** *ip-address* **source-interface** *if-type if-number* **traps version 2c** *name*
3. **snmp-server host** *ip-address* **source-interface** *if-type if-number* **use-vrf** *vrf-name*
4. **snmp-server host** *ip-address* **source-interface** *if-type if-number* [**udp_port** *number*]
5. **snmp-server source-interface** {**traps** | **informs**} *if-type if-number*
6. **show snmp source-interface**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server user** *name* [**auth** {**md5** | **sha**} *passphrase* [**auto**] [**priv** [**aes-128**] *passphrase*] [**engineID** *id*]
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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**} *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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*]

3. [**no**] **snmp-server host** *ip-address* **filter-vrf** *vrf-name* [**udp_port** *number*]
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server source-interface traps** *if-type if-number*
3. (Optional) **show snmp source-interface**
4. **snmp-server host** *ip-address* **use-vrf** *vrf-name* [**udp_port** *number*]
5. (Optional) **show snmp host**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
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 7: 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 remote monitoring (RMON) SNMP 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 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.
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.

Command	Purpose
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).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot*/*port*
3. **no snmp trap link-status**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

Displaying SNMP ifIndex for an Interface

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

SUMMARY STEPS

1. **show interface snmp-ifindex**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server tcp-session** [**auth**]
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server contact** *name*
3. **snmp-server location** *name*
4. (Optional) **show snmp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 3400 Series NX-OS Unicast Routing Configuration Guide.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]
3. (Optional) **snmp-server mib community-map** *community-name* **context** *context-name*
4. (Optional) **show snmp context**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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*] **Example:** `switch(config)# snmp-server context public1 vrf red`	Maps an SNMP context to a protocol instance, VRF, or topology. The names can be any alphanumeric string up to 32 characters. 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.

SUMMARY STEPS

1. **configure terminal**
2. **no snmp-server protocol enable**

DETAILED STEPS

	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.

Managing the SNMP Server Counter Cache Update Timer

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

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server counter cache timeout** *seconds*
3. (Optional) **show running-config snmp all |i cac**
4. **no snmp-server counter cache enable**

DETAILED STEPS

	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.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server aaa-user cache-timeout** *seconds*
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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

Configuring the SNMP Local Engine ID

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.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server engineID local** *engineid-string*
3. **show snmp engineID**
4. [**no**] **snmp-server engineID local** *engineid-string*
5. **copy running-config startup-config**

DETAILED STEPS

	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:** `switch(config)# no snmp-server engineID local AA:BB:CC:1A:2C:10`	Disables the local engine ID and the default auto-generated engine ID is configured.
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.
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.
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
```

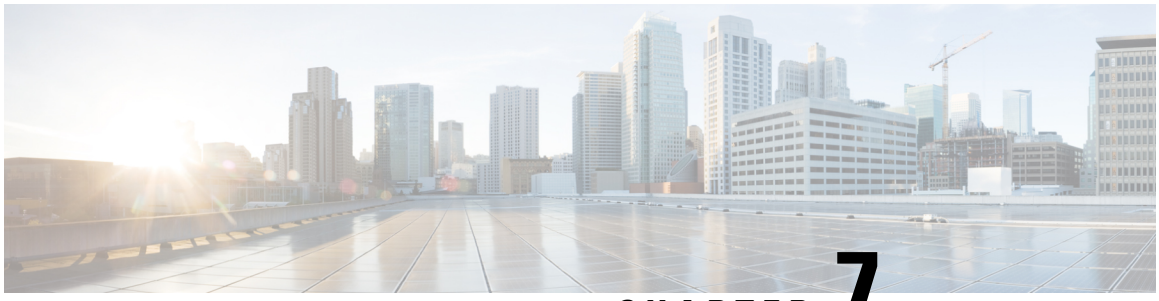
Additional References

Related Documents

Related Topic	Document Title
IP ACLs and AAA	*Cisco Nexus 3400 Series NX-OS Security Configuration Guide*

RFCs

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



CHAPTER 7

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 8: Bootup Diagnostics

Diagnostic	Description
Module	
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.
Supervisor	
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.
OBFL	Verifies the integrity of the onboard failure logging (OBFL) flash.

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 9: 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	30 minutes	active	Verifies the integrity of the primary boot device on a module.

Diagnostic	Default Interval	Default Setting	Description
Module			
SecondaryBootROM	30 minutes	active	Verifies the integrity of the secondary boot device on a module.
PortLoopback	24 hours	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.

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.

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.

Licensing Requirements for Online Diagnostics

Product	License Requirement
Cisco NX-OS	Online diagnostics require no license. Any feature not included in a license package is bundled with th image and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensin see the Cisco NX-OS Licensing Guide.

Guidelines and Limitations for Online Diagnostics

Online diagnostics has the following configuration guidelines and limitations:

- You cannot run disruptive online diagnostic tests on demand.
- The BootupPortLoopback test is not supported.
- 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.

SUMMARY STEPS

1. **configure terminal**
2. **diagnostic bootup level** {**complete** | **minimal** | **bypass**}
3. (Optional) **show diagnostic bootup level**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. **diagnostic monitor interval module** *slot* **test** [*test-id* | *name* | **all**] **hour** *hour* **min** *minute* **second** *second*
3. [**no**] **diagnostic monitor module** *slot* **test** [*test-id* | *name* | **all**]
4. (Optional) **show diagnostic content module** {*slot* | **all**}

DETAILED STEPS

	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.

	Command or Action	Purpose
		• *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.

SUMMARY STEPS

1. (Optional) **diagnostic ondemand iteration** *number*
2. (Optional) **diagnostic ondemand action-on-failure** {**continue failure-count** *num-fails* | **stop**}
3. **diagnostic start module** *slot* **test** [*test-id* | *name* | **all** | **non-disruptive**] [**port** *port-number* | **all**]
4. **diagnostic stop module** *slot* **test** [*test-id* | *name* | **all**]
5. (Optional) **show diagnostic status module** *slot*

DETAILED STEPS

	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.

SUMMARY STEPS

1. **diagnostic test simulation module** *slot* **test** *test-id* {**fail** | **random-fail** | **success**} [**port** *number* | **all**]

DETAILED STEPS

	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.

SUMMARY STEPS

1. **diagnostic clear result module** [*slot* | **all**] **test** {*test-id* | **all**}
2. **diagnostic test simulation module** *slot* **test** *test-id* clear

DETAILED STEPS

	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.
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.

Command	Purpose
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.

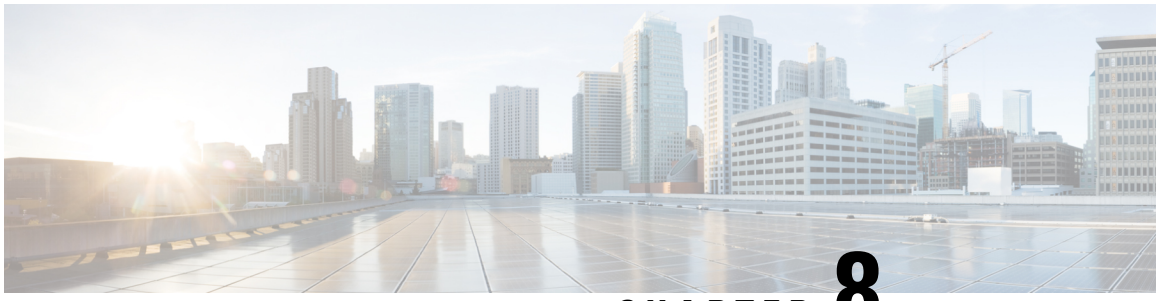
Configuration Examples for Online Diagnostics

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 8

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.

About EEM

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

EEM consists of three major components:

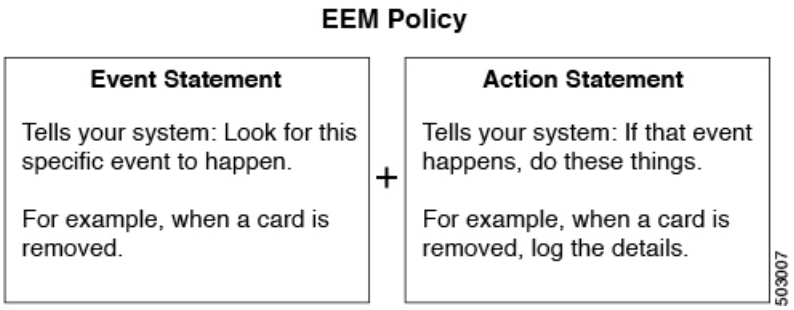
- Event statements—Events to monitor from another Cisco NX-OS component that may require some action, workaround, or notification.
- Action statements—An action that EEM can take, such as executing CLI commands, sending an email through the use of Smart Call Home feature, and disabling an interface to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

Policies

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

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

Figure 2: EEM Policy Statements



You can configure EEM policies using the command-line interface (CLI) or a VSH script.

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

EEM maintains event logs on the supervisor.

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

You can create user policies to suit your network. If you create a user policy, any actions in your policy occur after EEM triggers any system policy actions that are related to the same event as your policy.

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

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

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

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

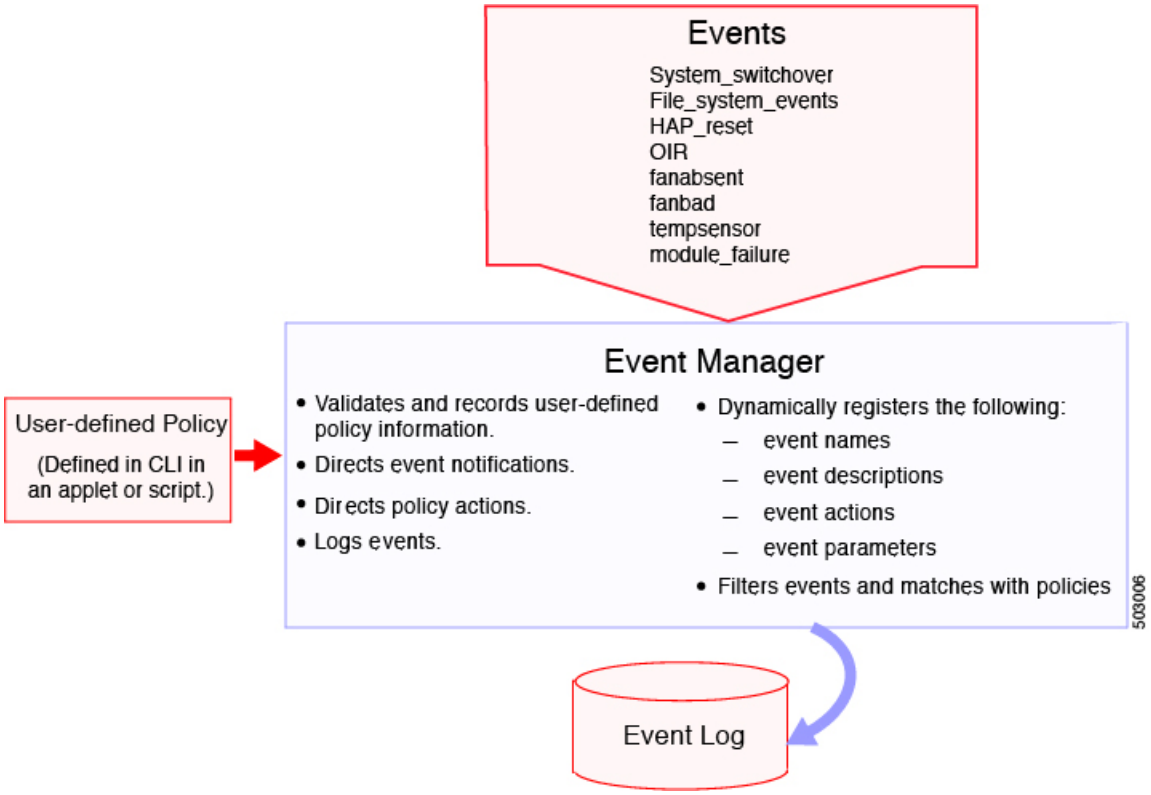
Event Statements

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

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

This figure shows events that are handled by EEM.

Figure 3: EEM Overview



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

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

Note If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the event default action statement.

Action Statements

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

EEM supports the following actions in action statements:

- Execute any CLI commands.
- Update a counter.
- Log an exception.
- Force the shutdown of any module.
- Reload the device.

- Shut down specified modules because the power is over budget.
- Generate a syslog message.
- Generate a Call Home event.
- Generate an SNMP notification.
- Use the default action for the system policy.

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

Note If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the default action. For example, if you match a CLI command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the CLI command to execute.

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

VSH Script Policies

You can also write policies in a VSH script, using a text editor. These policies have an event statement and action statement(s) just as other policies, and these policies can either augment or override system policies. After you write your VSH script policy, copy it to the device and activate it.

Environment Variables

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

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

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

```
switch (config-eem-policy)# action 1.0 forceshut module 1 reset-reson “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.

Licensing Requirements for EEM

Product	License Requirement
Cisco NX-OS	EEM requires no license. Any feature not included in a license package is bundled with the nx-os i provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing sche Cisco NX-OS Licensing Guide.

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 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 Python Scripting topic in the *Cisco Nexus 3400-S Series NX-OS Progammability 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.

SUMMARY STEPS

1. **configure terminal**

2. **event manager environment** *variable-name variable-value*
3. (Optional) **show event manager environment** {*variable-name* | **all**}
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**event manager environment** *variable-name variable-value* **Example:** `switch(config)# event manager environment emailto "admin@anyplace.com"`	Creates an environment variable for EEM. The *variable-name* can be any case-sensitive, alphanumeric string up to 29 characters. The *variable-value* can be any quoted alphanumeric string up to 39 characters.
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.

SUMMARY STEPS

1. **configure terminal**
2. **event manager applet** *applet-name*
3. (Optional) **description** *policy-description*
4. **event** *event-statement*
5. (Optional) **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*}
6. **action** *number*[*.number2*] *action-statement*
7. (Optional) **show event manager policy-state** *name* [**module** *module-id*]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 104.
Step 5	(Optional) **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*} **Example:** `switch(config-applet)# tag one or two happens 1 in 10000`	Correlates multiple events in the policy. The range for the *occurs* argument is from 1 to 4294967295. The range for the *seconds* argument is from 0 to 4294967295 seconds.
Step 6	**action** *number*[.*number2*] *action-statement* **Example:** `switch(config-applet)# action 1.0 cli show interface e 3/1`	Configures an action statement for the policy. Repeat this step for multiple action statements. See Configuring Action Statements, on page 109.
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.
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.

Command	Purpose
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.
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 113.
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.

Command	Purpose
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.
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.

Command	Purpose
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.
event syslog [**occurs** *count*] {**pattern** *string* | **period** *time* | **priority** *level* | **tag** *tag*} **Example:** `switch(config-applet)# event syslog period 500`	Triggers an event if the specified syslog threshold is exceeded. The range for the count is from 1 to 65000, and the range for the time is from 1 to 4294967295. The priority range is from 0 to 7. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy.
event sysmgr memory [**module** *module-num*] **major** *major-percent* **minor** *minor-percent* **clear** *clear-percent* **Example:** `switch(config-applet)# event sysmgr memory minor 80`	Triggers an event if the specified system manager memory threshold is exceeded. The range for the percentage is from 1 to 99.
event sysmgr switchover count *count* **time** *interval* **Example:** `switch(config-applet)# event sysmgr switchover count 10 time 1000`	Triggers an event if the specified switchover count is exceeded within the time interval specified. The switchover count is from 1 to 65000. The time interval is from 0 to 2147483647.
event temperature [**module** *slot*] [*sensor-number*] **threshold** {**any** | **major** | **minor**} **Example:** `switch(config-applet)# event temperature module 2 threshold any`	Triggers an event if the temperature sensor exceeds the configured threshold. The sensor range is from 1 to 18.

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

Configuring Action Statements

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

Command	Purpose
action *number*[*.number2*] **cli** *command1* [*command2...*] [**local**] **Example:** `switch(config-applet)# action 1.0 cli "show interface e 3/1"`	Runs the configured CLI commands. You can optionally run the commands on the module where the event occurred. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.

Command	Purpose
action *number*[.*number2*] **counter name** *counter* **value** *val* **op** {**dec** | **inc** | **nop** | **set**} **Example:** `switch(config-applet)# action 2.0 counter name mycounter value 20 op inc`	Modifies the counter by the configured value and operation. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The counter name can be any case-sensitive, alphanumeric string up to 28 characters. The *val* can be an integer from 0 to 2147483647 or a substituted parameter.
action *number*[.*number2*] **event-default** **Example:** `switch(config-applet)# action 1.0 event-default`	Executes the default action for the associated event. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[.*number2*] **forceshut** [**module** *slot* | **xbar** *xbar-number*] **reset-reason** *seconds* **Example:** `switch(config-applet)# action 1.0 forceshut module 2 reset-reason "flapping links"`	Forces a module, crossbar, or the entire system to shut down. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The reset reason is a quoted alphanumeric string up to 80 characters.
action *number*[.*number2*] **overbudgetshut** [**module** *slot*[-*slot*]] **Example:** `switch(config-applet)# action 1.0 overbudgetshut module 3-5`	Forces one or more modules or the entire system to shut down because of a power overbudget issue. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[.*number2*] **policy-default** **Example:** `switch(config-applet)# action 1.0 policy-default`	Executes the default action for the policy that you are overriding. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[.*number2*] **publish-event** **Example:** `switch(config-applet)# action 1.0 publish-event`	Forces the publication of an application-specific event. The action label is in the format number1.number2. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[.*number2*] **reload** [**module** *slot*[-*slot*]] **Example:** `switch(config-applet)# action 1.0 reload module 3-5`	Forces one or more modules or the entire system to reload. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.

Command	Purpose
action *number*[*.number2*] **snmp-trap** {[**intdata1** *data* [**intdata2** *data*]] [**strdata** *string*]} **Example:** `switch(config-applet)# action 1.0 snmp-trap strdata "temperature problem"`	Sends an SNMP trap with the configured data. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *data* arguments can by any number up to 80 digits. The *string* can be any alphanumeric string up to 80 characters.
action *number*[*.number2*] **syslog** [**priority** *prio-val*] **msg** *error-message* **Example:** `switch(config-applet)# action 1.0 syslog priority notifications msg "cpu high"`	Sends a customized syslog message at the configured priority. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *error-message* can be any quoted alphanumeric string up to 80 characters.

Note If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the default action. For example, if you match a CLI command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the CLI command to execute. You can use the **terminal event-manager bypass** command to allow all EEM policies with CLI matches to execute the CLI command.

Defining a Policy Using a VSH Script

You can define a policy using a VSH script.

Before you begin

Ensure that you are logged in with administrator privileges.

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

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.

SUMMARY STEPS

1. **configure terminal**
2. **event manager policy** *policy-script*
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show event manager policy-state** *system-policy*
3. **event manager applet** *applet-name* **override** *system-policy*
4. (Optional) **description** *policy-description*
5. **event** *event-statement*
6. **action** *number action-statement*
7. (Optional) **show event manager policy-state** *name*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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`
		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. For information about system policies, see Embedded Event Manager System Events and Configuration Examples, on page 217.

	Command or Action	Purpose
	`Cfg time interval : 10.000000 (seconds)` `Hash default, Count 0`	

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.

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.

SUMMARY STEPS

1. **configure terminal**
2. **system memory-thresholds minor** *minor* **severe** *severe* **critical** *critical*
3. (Optional) **system memory-thresholds threshold critical no-process-kill**
4. (Optional) **show running-config | include "system memory"**

5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

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"**	Displays information about the system memory configuration.

	Command or Action	Purpose
	Example: `switch(config-applet)# show running-config | include "system memory"`	
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.

SUMMARY STEPS

1. **configure terminal**
2. **event manager applet** *applet-name*
3. **event syslog** [**tag** *tag*] {**occurs** *number* | **period** *seconds* | **pattern** *msg-text* | **priority** *priority*}
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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:**	Monitors syslog messages and invokes the policy based on the search string in the policy.

	Command or Action	Purpose
	`switch(config-applet)# event syslog occurs 10`	• 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. • 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.

Command	Purpose
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
```

Upon reaching a maximum failure threshold, the AsicMemory, FpgaRegTest, and L2ACLRedirect system policies force a reload of the switch. This example shows how to override the default action for one of these policies and issue a syslog instead:

```
event manager applet gold override __fpgareg
action 1 syslog priority emergencies msg FpgaRegTest_override
```

This example shows how to override a default policy but still enact the default action:

```
event manager applet gold_fpga_ovrd override __fpgareg
  action 1 policy-default
  action 2 syslog priority emergencies msg FpgaRegTest_override
```

Note For additional EEM configuration examples, see Embedded Event Manager System Events and Configuration Examples, on page 217.

Event Log Auto-Collection and Backup

Automatically collected event logs are stored locally on switch memory. Event log file storage is a temporary buffer that stores files for a fixed amount of time. Once the time period has elapsed, a roll-over of the buffer makes room for the next files. The roll-over uses a first-in-first-out method.

Beginning with Cisco NX-OS Release 9.3(3), EEM uses the following methods of collection and backup:

- Extended Log File Retention
- Trigger-Based Event Log Auto-Collection

Extended Log File Retention

Beginning with Cisco NX-OS release 9.3(3), all Cisco Nexus platform switches, with at least 8Gb of system memory, support the extended retention of event logging files. Storing the log files locally on the switch or remotely through an external container, reduces the loss of event logs due to rollover.

Enabling Extended Log File Retention For All Services

copied updates from N9K version

SUMMARY STEPS

1. **configure terminal**
2. **bloggerd log-dump all**

DETAILED STEPS

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

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

Step 2	**bloggerd log-dump all**
	Example:
	`switch(config)# bloggerd log-dump all`
	`switch(config)#`
		Enables the log file retention feature for all services.

Example

```
switch# configure terminal
switch(config)# bloggerd log-dump all
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)#
```

Disabling Extended Log File Retention For All Services

Extended Log File Retention is disabled by default for all services on the switch. If the switch has the log file retention feature enabled for all services and you want to disable it, use the following procedure.

SUMMARY STEPS

1. **configure terminal**
2. **no bloggerd log-dump all**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**no bloggerd log-dump all**
	Example:
	`switch(config)# no bloggerd log-dump all`
	`switch(config)#`
		Disables the log file retention feature for all services on the switch.

Example

```
switch# configure terminal
switch(config)# no bloggerd log-dump all
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)#
```

Enabling Extended Log File Retention For a Single Service

Extended Log File Retention is enabled by default for all services running on a switch. If the switch doesn't have the log file retention feature enabled (**no bloggerd log-dump** is configured), use the following procedure to enable it for a single service.

SUMMARY STEPS

1. **show system internal sysmgr service name** *service-type*
2. **configure terminal**
3. **bloggerd log-dump sap** *number*
4. **show system internal bloggerd info log-dump-info**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show system internal sysmgr service name** *service-type* **Example:** `switch# show system internal sysmgr service name aclmgr`	Displays information about the ACL Manager including the service SAP number.

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

Step 3	**bloggerd log-dump sap** *number* **Example:** `switch(config)# bloggerd log-dump sap 351`	Enables the log file retention feature for the ACL Manager service.
Step 4	**show system internal bloggerd info log-dump-info** **Example:** `switch(config)# show system internal bloggerd info log-dump-info`	Displays information about the log file retention feature on the switch.

Example

```
switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
        UUID = 0x182, PID = 653, SAP = 351
        State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
        Restart count: 1
        Time of last restart: Mon Nov  4 11:10:39 2019.
        The service never crashed since the last reboot.
        Tag = N/A
        Plugin ID: 0
switch(config)# configure terminal
switch(config)# bloggerd log-dump sap 351
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
```

```
switch(config)# show system internal bloggerd info log-dump-info
------------------------------------------------------------
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
------------------------------------------------------------
Module     | VDC        | SAP                        | Enabled?
------------------------------------------------------------
1          | 1          | 351 (MTS_SAP_ACLMGR     ) | Enabled
------------------------------------------------------------
------------------------------------------------------------
Log Dump Throttle Switch-Wide Config:
-------------------------------------
Log Dump Throttle                                    : ENABLED
Minimum buffer rollover count (before throttling)    : 5
Maximum allowed rollover count per minute            : 1
------------------------------------------------------------

switch(config)#
```

Displaying Extended Log Files

Use this task to display the event log files currently stored on the switch.

SUMMARY STEPS

1. **dir debug:log-dump/**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**dir debug:log-dump/** **Example:** `switch# dir debug:log-dump/`	Displays the event log files currently stored on the switch.

Example

```
switch# dir debug:log-dump/

3676160 Dec 05 02:43:01 2019 20191205023755_evtlog_archive.tar
3553280 Dec 05 06:05:06 2019 20191205060005_evtlog_archive.tar

Usage for debug://sup-local
913408 bytes used
4329472 bytes free
5242880 bytes total
```

Disabling Extended Log File Retention For a Single Service

Extended Log File Retention is enabled by default for all services on the switch. If the switch has the log file retention feature enabled for a single service or all services (by default in Cisco NX-OS Release 9.3(5)), and you want to disable a specific service or services, use the following procedure.

SUMMARY STEPS

1. **show system internal sysmgr service name** *service-type*
2. **configure terminal**
3. **no bloggerd log-dump sap** *number*
4. **show system internal bloggerd info log-dump-info**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show system internal sysmgr service name** *service-type* **Example:** `switch# show system internal sysmgr service name aclmgr`	Displays information about the ACL Manager including the service SAP number.

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

Step 3	**no bloggerd log-dump sap** *number* **Example:** `switch(config)# no bloggerd log-dump sap 351`	Disables the log file retention feature for the ACL Manager service.
Step 4	**show system internal bloggerd info log-dump-info** **Example:** `switch(config)# show system internal bloggerd info log-dump-info`	Displays information about the log file retention feature on the switch.

Example

The following example shows how to disable extended log file retention for a service named "aclmgr":

```
switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
        UUID = 0x182, PID = 653, SAP = 351
        State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
        Restart count: 1
        Time of last restart: Mon Nov  4 11:10:39 2019.
        The service never crashed since the last reboot.
        Tag = N/A
        Plugin ID: 0
switch(config)# configure terminal
switch(config)# no bloggerd log-dump sap 351
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)# show system internal bloggerd info log-dump-info
------------------------------------------------------------
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
------------------------------------------------------------
```

```
Module     | VDC        | SAP                         | Enabled?
-------------------------------------------------------------
1          | 1          | 351 (MTS_SAP_ACLMGR     ) | Disabled
-------------------------------------------------------------
-------------------------------------------------------------
Log Dump Throttle Switch-Wide Config:
-------------------------------------
Log Dump Throttle                                    : ENABLED
Minimum buffer rollover count (before throttling)    : 5
Maximum allowed rollover count per minute            : 1
-------------------------------------------------------------

switch(config)#
```

Trigger-Based Event Log Auto-Collection

Trigger-based log collection capabilities:

- Automatically collect relevant data when issues occur.
- No impact on control plane
- Customizable configuration:
  - Defaults populated by Cisco
  - Selectively override what-to-collect by network administrator or by Cisco TAC.
  - Automatically update new triggers on image upgrades.
- Store logs locally on the switch or remotely on an external server.
- Supports severity 0, 1, and 2 syslogs:
- Custom syslogs for ad-hoc events (auto-collection commands attached to the syslogs)

Enabling Trigger-Based Log File Auto-Collection

To enable trigger-based automatic creation of log files, you must create an override policy for the __syslog_trigger_default system policy with a custom YAML file and define the specific logs for which information will be collected.

For more information on creating a custom YAML file to enable log file auto-collection, see Configuring the Auto-Collection YAML File, on page 124.

Auto-Collection YAML File

The Auto-Collection YAML file that is specified in the **action** command in the EEM function, defines actions for different system or feature components. This file is located in the switch directory: /bootflash/scripts. In addition to the default YAML file, you can create component-specific YAML files and place them in the same directory. The naming convention for component-specific YAML files is **component-name.yaml**. If a component-specific file is present in the same directory, it takes precedence over the file that is specified in the **action** command. For example, if the action file, **bootflash/scripts/platform.yaml** is in the **/bootflash/scripts** directory with the default action file, **bootflash/scripts/test.yaml**, then the instructions defined in **platform.yaml** file take precedence over the instructions for the platform component present in the default **test.yaml** file.

Examples of components are, ARP, BGP, IS-IS, and so on. If you are not familiar with all the component names, contact Cisco Customer Support for assistance in defining the YAML file for component-specific actions (and for the default **test.yaml** file as well).

Example:

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

Configuring the Auto-Collection YAML File

A contents of a YAML file determines the data collected during trigger-based auto-collection. There must be only one YAML file on the switch but it can contain auto-collection meta-data for any number of switch components and messages.

Locate the YAML file in the following directory on the switch:

```
/bootflash/scripts
```

Invoke the YAML file for trigger-based collection by using the following example. The example shows the minimum required configuration for trigger-based collection to work with a user-defined YAML file.

```
switch# show running-config eem
!Command: show running-config eem
!Running configuration last done at: Mon Sep 30 19:34:54 2019
!Time: Mon Sep 30 22:24:55 2019
version 9.3(3) Bios:version 07.59
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

In the preceding example, "`test_1`" is the name of the applet and "`test.yaml`" is the name of the user-configured YAML file present in the /bootflash/scripts directory.

Example YAML File

The following is an example of a basic YAML file supporting the trigger-based event log auto-collection feature. The definitions for the keys/values in the file are in the table that follows.

Note Make sure that the YMAL file has proper indentation. As a best practice, run it through any "online YAML validator" before using it on a switch.

```
bash-4.3$ cat /bootflash/scripts/test.yaml
version: 1
components:
   securityd:
        default:
            tech-sup: port
            commands: show module
   platform:
        default:
            tech-sup: port
            commands: show module
```

Key: Value	Description
version: 1	Set to 1. Any other number creates an incompatibility for the auto collect script.
components:	Keyword specifying that what follows are switch components.

Key: Value	Description
securityd:	Name of the syslog component (`securityd` is a facility name in syslog).
default:	Identifies all messages belonging to the component.
tech-sup: port	Collect tech support of the port module for the `securityd` syslog component.
commands: show module	Collect show module command output for the `securityd` syslog component.
platform:	Name of the syslog component (`platform` is a facility name in syslog).
tech-sup: port	Collect tech support of the port module for the `platform` syslog component.
commands: show module	Collect show module command output for the `platform` syslog component.

Use the following example to associate auto-collect metadata only for a specific log. For example, SECURITYD-2-FEATURE_ENABLE_DISABLE

```
securityd:
                feature_enable_disable:
                        tech-sup: security
                        commands: show module
```

Key: Value	Description
securityd:	Name of the syslog component (`securityd` is a facility name in syslog).
feature_enable_disable:	Message ID of the syslog message.
tech-sup: security	Collect tech support of the security module for the `securityd` syslog component.
commands: show module	Collect show module command output for the security syslog component.

Example syslog output for the above YAML entry:

```
2019 Dec 4 12:41:01 n9k-c93108tc-fx %SECURITYD-2-FEATURE_ENABLE_DISABLE: User
has enabled the feature bash-shell
```

Use the following example to specify multiple values.

```
version: 1
components:
   securityd:
        default:
            commands: show module;show version;show module
            tech-sup: port;lldp
```

Note Use semicolons to separate multiple show commands and tech support key values (see the preceding example).

Limiting the Amount of Auto-Collections Per Component

For auto-collection, the limit of the number of bundles per component event is set to three (3) by default. If more than three events occur for a component, then the events are dropped with the status message **EVENTLOGLIMITREACHED**. The auto-collection of the component event restarts when the event log has rolled over.

Example:

```
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog                   Status/Secs/Logsize(Bytes)
2020-Jun-27 07:20:03  1140276903   ACLMGR-0-TEST_SYSLOG     EVENTLOGLIMITREACHED
2020-Jun-27 07:15:14  1026359228   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:15:09  384952880    ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:13:55  1679333688   ACLMGR-0-TEST_SYSLOG     PROCESSED:2:9332278
2020-Jun-27 07:13:52  1679333688   ACLMGR-0-TEST_SYSLOG     PROCESSING
2020-Jun-27 07:12:55  502545693    ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:12:25  1718497217   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:08:25  1432687513   ACLMGR-0-TEST_SYSLOG     PROCESSED:2:10453823
2020-Jun-27 07:08:22  1432687513   ACLMGR-0-TEST_SYSLOG     PROCESSING
2020-Jun-27 07:06:16  90042807     ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:03:26  1737578642   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:02:56  40101277     ACLMGR-0-TEST_SYSLOG     PROCESSED:3:10542045
2020-Jun-27 07:02:52  40101277     ACLMGR-0-TEST_SYSLOG     PROCESSING
```

Auto-Collection Log Files

About Auto-Collection Log Files

The configuration in a YAML file determines the contents of an auto-collected log file. You can't configure the amount of memory used for collected log files. You can configure the frequency of when the stored files get purged.

Autocollected log files get saved in the following directory:

```
switch# dir bootflash:eem_snapshots
   44205843    Sep 25 11:08:04 2019
1480625546_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz
  Usage for bootflash://sup-local
 6940545024 bytes used
44829761536 bytes free
51770306560 bytes total
```

Accessing the Log Files

Locate the logs by using the command keyword "`debug`":

```
switch# dir debug:///
...
       26    Oct 22 10:46:31 2019  log-dump
       24    Oct 22 10:46:31 2019  log-snapshot-auto
       26    Oct 22 10:46:31 2019  log-snapshot-user
```

The following table describes the log locations and the log types stored.

Location	Description
log-dump	This folder stores Event logs on log rollover.
log-snapshot-auto	This folder contains the auto-collected logs for syslog events 0, 1, 2.

Location	Description
log-snapshot-user	This folder stores the collected logs when you run the `bloggerd log-snapshot <>` command.

Use the following example to view the log files generated on log rollover:

```
switch# dir debug:log-dump/
debug:log-dump/20191022104656_evtlog_archive.tar
debug:log-dump/20191022111241_evtlog_archive.tar
debug:log-dump/20191022111841_evtlog_archive.tar
debug:log-dump/20191022112431_evtlog_archive.tar
debug:log-dump/20191022113042_evtlog_archive.tar
debug:log-dump/20191022113603_evtlog_archive.tar
```

Parsing the Log tar Files

Use the following example to parse the logs in the tar files:

```
switch# show system internal event-logs parse debug:log-dump/20191022104656_evtlog_archive.tar
--------LOGS:/tmp/BLOGGERD0.991453012199/tmp/1-191022104658-191022110741-device_test-M27-V1-I1:0-P884.gz--------
2019 Oct 22 11:07:41.597864 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Data Space
Limits(bytes): Soft: -1  Ha rd: -1
2019 Oct 22 11:07:41.597857 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Stack Space
Limits(bytes): Soft: 500000  Hard: 500000
2019 Oct 22 11:07:41.597850 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):AS: 1005952076
-1
2019 Oct 22 11:07:41.597406 E_DEBUG Oct 22 11:07:41 2019(device_test_process_events):Sdwrap
 msg unknown
2019 Oct 22 11:07:41.597398 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Going back to
select
2019 Oct 22 11:07:41.597395 E_DEBUG Oct 22 11:07:41 2019(nvram_test):TestNvram examine 27
blocks
2019 Oct 22 11:07:41.597371 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Parent: Thread
created test index:4 thread_id:-707265728
2019 Oct 22 11:07:41.597333 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Node inserted
2019 Oct 22 11:07:41.597328 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):The test index
in diag is 4
2019 Oct 22 11:07:41.597322 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):result severity
level
2019 Oct 22 11:07:41.597316 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):callhome alert
level
```

The following table describes the additional keywords available for parsing the specific tar file:

Keyword	Description
component	Decode logs belonging to the component identified by process name.
from-datetime	Decode logs from a specific date and time in yy[mm[dd[HH[MM[SS]]]]] format.
instance	List of SDWRAP buffer instances to be decoded (comma separated).
module	Decode logs from modules such as SUP and LC (using module IDs).
to-datetime	Decode logs up to a specific date and time in yy[mm[dd[HH[MM[SS]]]]] format.

Copying Logs to a Different Location

Use the following example to copy logs to a different location such as a remote server:

```
switch# copy debug:log-dump/20191022104656_evtlog_archive.tar
scp://<ip-adress>/nobackup/<user> vrf management  use-kstack
Enter username: user@<ip-address>'s password:
20191022104656_evtlog_archive.tar                                          100%  130KB
130.0KB/s   00:00
Copy complete, now saving to disk (please wait)...
Copy complete.
```

Purging Auto-Collection Log Files

There are two types of generated trigger-based auto-collection logs: EventHistory and EventBundle.

Purge Logic for EventHistory Logs

For event history, purging occurs in the /var/sysmgr/srv_logs/xport folder. 250MB of partitioned RAM is mounted at /var/sysmgr/srv_logs directory.

If the /var/sysmgr/srv_logs memory usage is under 65% of the 250MB allocated, no files get purged. When the memory utilization reaches the 65% limit level, the oldest files get purged until there's enough memory available to continue saving new logs.

Purge Logic for EventBundle Logs

For event bundles, the purge logic occurs in the /bootflash/eem_snapshots folder. For storing the auto-collected snapshots, the EEM auto-collect script allocates 5% of the bootflash storage. The logs get purged once the 5% bootflash capacity is used.

When a new auto-collected log is available but there's no space to save it in bootflash (already at 5% capacity), the system checks the following:

1. If there are existing auto-collected files that are more than 12 hours old, the system deletes the files and the new logs get copied.
2. If the existing auto collected files are less than 12 hours old, the system discards the newly collected logs without saving them.

You can modify the 12-hour default purge time by using the following commands. The time specified in the command is in minutes.

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml purge-time 300 $_syslog_msg
```

event manager command: *test* is an example name for the policy. **__syslog_trigger_default** is the name of the system policy that you want to override. This name must begin with a double underscore (__).

action command: **1.0** is an example number for the order in which the action is executed. **collect** indicates that data is collected using the YAML file. *test.yaml* is an example name of the YAML file. **$_syslog_msg** is the name of the component.

Note At any given time, there can be only one trigger-based auto-collection event in progress. If another new log event is attempting to be stored when auto-collection is already occurring, the new log event is discarded.

By default, there's only one trigger-based bundle collected every five minutes (300 sec). This rate limiting is also configurable by the following commands. The time specified in the command is in seconds.

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml rate-limit 600 $_syslog_msg
```

event manager command: *test* is an example name for the policy. **__syslog_trigger_default** is an example name of the system policy to override. This name must begin with a double underscore (__).

action command: **1.0** is an example number for the order in which the action is executed. **collect** indicates that data is collected using the YAML file. *test.yaml* is an example name of the YAML file. **$_syslog_msg** is the name of the component.

Auto-Collection Statistics and History

The following example shows trigger-based collection statistics:

```
switch# show system internal event-logs auto-collect statistics
--------------------EEM Auto Collection Statistics--------------------
Syslog Parse Successful :88 Syslog Parse Failure :0
Syslog Ratelimited :0 Rate Limit Check Failed :0
Syslog Dropped(Last Action In Prog) :53 Storage Limit Reached :0
User Yaml Action File Unavailable :0 User Yaml Parse Successful :35
User Yaml Parse Error :0 Sys Yaml Action File Unavailable :11
Sys Yaml Parse Successful :3 Sys Yaml Parse Error :0
Yaml Action Not Defined :0 Syslog Processing Initiated :24
Log Collection Failed :0 Tar Creation Error :0
Signal Interrupt :0 Script Exception :0
Syslog Processed Successfully :24 Logfiles Purged :0
```

The following example shows trigger-based collection history (the processed syslogs, process time, size of the data collected) obtained using a CLI command:

```
switch# show system internal event-logs auto-collect history
DateTime Snapshot ID Syslog Status/Secs/Logsize(Bytes)
2019-Dec-04 05:30:32 1310232084 VPC-0-TEST_SYSLOG PROCESSED:9:22312929
2019-Dec-04 05:30:22 1310232084 VPC-0-TEST_SYSLOG PROCESSING
2019-Dec-04 04:30:13 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSED:173:33194665
2019-Dec-04 04:28:47 897805674 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:28:47 947981421 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:27:19 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSING
2019-Dec-04 02:17:16 1957148102 CARDCLIENT-2-FPGA_BOOT_GOLDEN NOYAMLFILEFOUND
```

Verifying Trigger-Based Log Collection

Verify that the trigger-based log collection feature is enabled by entering the **show event manager system-policy | i trigger** command as in this example:

```
switch# show event manager system-policy | i trigger n 2
           Name : __syslog_trigger_default
    Description : Default policy for trigger based logging
    Overridable : Yes
     Event type : 0x2101
```

Checking Trigger-Based Log File Generation

You can check to see if the trigger-based auto-collection feature has generated any event log files. Enter one of the commands in the following examples:

```
switch# dir bootflash:eem_snapshots
9162547 Nov 12 22:33:15 2019 1006309316_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz

Usage for bootflash://sup-local
8911929344 bytes used
3555950592 bytes free
12467879936 bytes total
```

```
switch# dir debug:log-snapshot-auto/
63435992 Dec 03 06:28:52 2019
20191203062841_1394408030_PLATFORM_2_MOD_PWRDN_eem_snapshot.tar.gz

Usage for debug://sup-local
544768 bytes used
4698112 bytes free
5242880 bytes total
```

Local Log File Storage

Local log file storage capabilities:

- Amount of local data storage time depends on the scale, and type, of deployment. For both modular and nonmodular switches, the storage time is from 15 minutes to several hours of data. To be able to collect relevant logs that span a longer period:
  - Only enable event log retention for the specific services/features you need. See Enabling Extended Log File Retention For a Single Service , on page 120.
  - Export the internal event logs off the switch. See External Log File Storage, on page 132.
- Compressed logs are stored in RAM.
- 250MB memory is reserved for log file storage.
- Log files are optimized in tar format (one file for every five minutes or 10MB, whichever occurs first).
- Allow snap-shot collection.

Generating a Local Copy of Recent Log Files

Extended Log File Retention is enabled by default for all services running on a switch. For local storage, the log files are stored on flash memory. Use the following procedure to generate a copy of up to ten of the most recent event log files.

SUMMARY STEPS

1. **bloggerd log-snapshot** [*file-name*] [**bootflash:** *file-path* | **logflash:** *file-path* | **usb1:**] [**size** *file-size*] [**time** *minutes*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**bloggerd log-snapshot** [*file-name*] [**bootflash:** *file-path* | **logflash:** *file-path* | **usb1:**] [**size** *file-size*] [**time** *minutes*] **Example:** `switch# bloggerd log-snapshot snapshot1`	Creates a snapshot bundle file of the last ten event logs stored on the switch. Default storage for this operation is **logflash**. *file-name*: The filename of the generated snapshot log file bundle. Use a maximum of 64 characters for *file-name*.

	Command or Action	Purpose
		Note This variable is optional. If it is not configured, the system applies a timestamp and "_snapshot_bundle.tar" as the filename. Example: `20200605161704_snapshot_bundle.tar` **bootflash:** *file-path*: The file path where the snapshot log file bundle is being stored on the bootflash. Choose one of the following initial paths: • bootflash:/// • bootflash://module-1/ • bootflash://sup-1/ • bootflash://sup-active/ • bootflash://sup-local/ **logflash:** *file-path*: The file path where the snapshot log file bundle is being stored on the logflash. Choose one of the following initial paths: • logflash:/// • logflash://module-1/ • logflash://sup-1/ • logflash://sup-active/ • logflash://sup-local/ **usb1:**: The file path where the snapshot log file bundle is being stored on the USB device. **size** *file-size*: The snapshot log file bundle based on size in megabytes (MB). Range is from 5MB through 250MB. **time** *minutes*: The snapshot log file bundle based on the last x amount of time (minutes). Range is from 1 minute through 30 minutes.

Example

```
switch# bloggerd log-snapshot snapshot1
Snapshot generated at logflash:evt_log_snapshot/snapshot1_snapshot_bundle.tar Please cleanup
 once done.
switch#
switch# dir logflash:evt_log_snapshot
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for logflash://sup-local
```

```
759865344 bytes used
5697142784 bytes free
6457008128 bytes total
```

Display the same files using the command in this example:

```
switch# dir debug:log-snapshot-user/
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for debug://sup-local
929792 bytes used
4313088 bytes free
5242880 bytes total
```

Note The file name is identified at the end of the example. Each individual log file is also identified by the date and time it was generated.

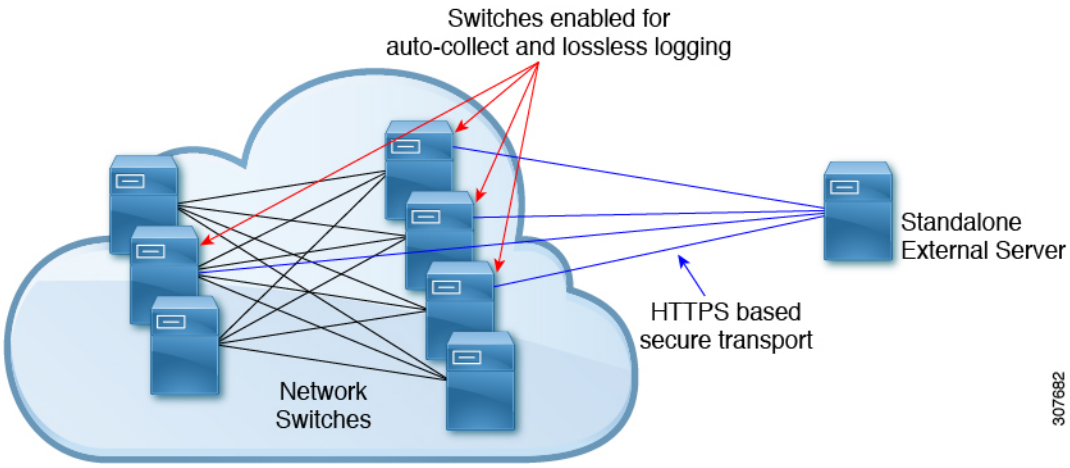
External Log File Storage

An external server solution provides the capability to store logs off-switch in a secure manner.

Note To create the external storage capability, contact Cisco Technical Assistance Center(TAC) to help deploy the external server solution.

Te following are external log file storage capabilities:

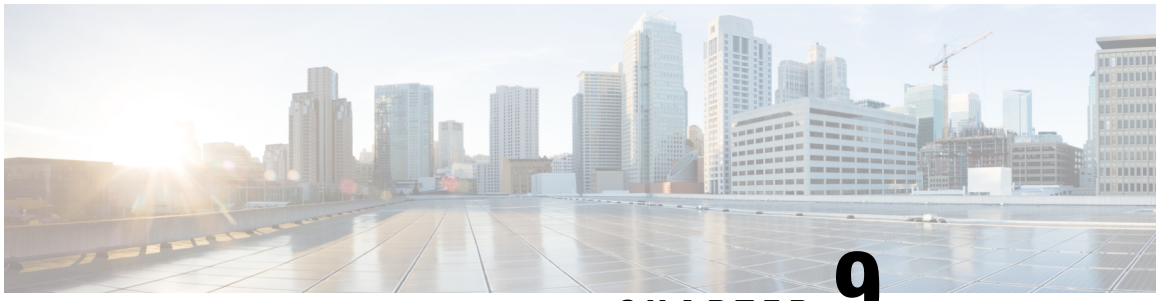
- Enabled on-demand
- HTTPS-based transport
- Storage requirements:
  - Nonmodular switches: 300MB
  - Modular switches: 12GB (per day, per switch)
- An external server generally stores logs for 10 switches. However, there's no firm limit to the number of switches supported by an external server.



The external server solution has the following characteristics:

- Controller-less environment
- Manual management of security certificates
- Three supported use-cases:
  - Continuous collection of logs from selected switches
  - TAC-assisted effort to deploy and upload logs to Cisco servers.
  - Limited on-premise processing

Note Contact Cisco TAC for information regarding the setup and collection of log files in an external server.



CHAPTER 9

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

Licensing Requirements for OBFL

Product	License Requirement
Cisco NX-OS	OBFL requires no license. Any feature not included in a license package is bundled with the nx-os im provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, Cisco NX-OS Licensing Guide.

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.

SUMMARY STEPS

1. **configure terminal**
2. **hw-module logging onboard**
3. **hw-module logging onboard counter-stats**
4. **hw-module logging onboard cpuhog**
5. **hw-module logging onboard environmental-history**
6. **hw-module logging onboard error-stats**
7. **hw-module logging onboard interrupt-stats**
8. **hw-module logging onboard module** *slot*
9. **hw-module logging onboard obfl-logs**
10. (Optional) **show logging onboard**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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**	Enables the OBFL CPU hog events.

	Command or Action	Purpose

	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.`	

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**
	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.`
		Enables the OBFL error statistics.

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**	Displays information about OBFL.

	Command or Action	Purpose
	Example: `switch(config)# show logging onboard`	**Note** To display OBFL information stored in flash on a module, see Verifying the OBFL Configuration, on page 139
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the OBFL Configuration

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

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

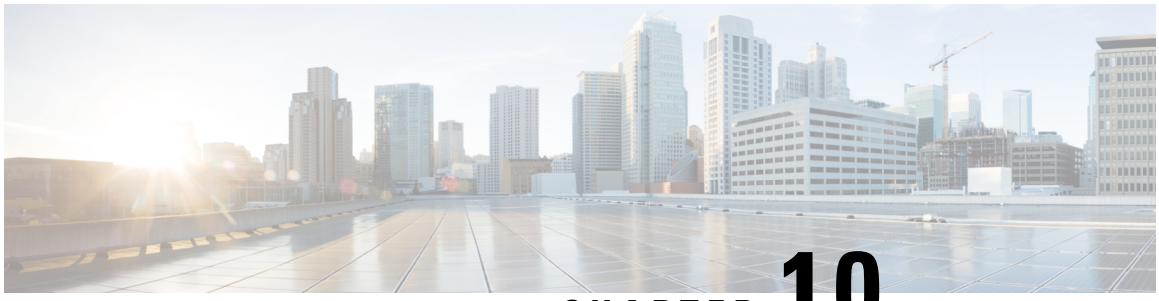
This example shows how to enable OBFL on module 2 for environmental information:

```
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 3400 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 10

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 includes 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)
- Port channels

Characteristics of Source Ports

SPAN source ports have the following characteristics:

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

SPAN Destinations

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

- Ethernet ports in either access or trunk mode

Characteristics of Destination Ports

SPAN destination ports have the following characteristics:

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

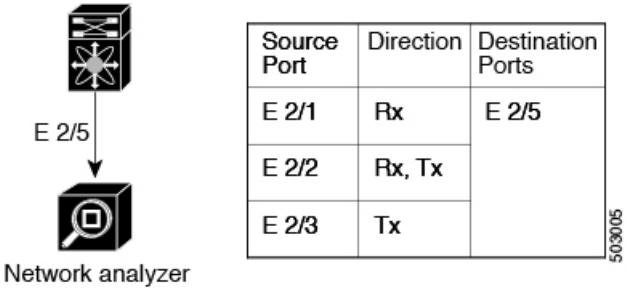
SPAN Sessions

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

See the *Cisco Nexus 3400-S Series NX-OS Verified Scalability Guide* for information on the number of supported SPAN sessions.

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

Figure 4: SPAN Configuration



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 3400-S Series NX-OS Security Configuration Guide*.

Licensing Requirements for SPAN

Product	License Requirement
Cisco NX-OS	SPAN requires no license. Any feature not included in a license package is bundled with the nx-os provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing sche Cisco NX-OS Licensing 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 3400-S Series NX-OS Interfaces Configuration Guide*.

Guidelines and Limitations for SPAN

Note For scale information, see the release-specific *Cisco Nexus 3400-S NX-OS Verified Scalability Guide*.

SPAN has the following configuration guidelines and limitations:

- All SPAN replication is performed in the hardware. The supervisor CPU is not involved.
- In SPAN sessions, destination as a Port channel is not supported.
- 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.
- For SPAN session limits, see the *Cisco Nexus 3400-s Series NX-OS Verified Scalability Guide*.
- You can configure only one destination port in a SPAN session.
- Interfaces configured as part of one SPAN/ERSPAN session as source interfaces cannot be used in other SPAN/ERSPAN sessions.
- A destination port can be configured in only one SPAN session at a time.
- 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 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.
- When a SPAN session contains source ports that are monitored in the transmit or transmit and receive direction, packets that these ports receive might be replicated to the SPAN destination port even though the packets are not actually transmitted on the source ports. Some examples of this behavior on source ports are as follows:
  - Traffic that results from flooding
  - Broadcast traffic
- 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.
- When using **shut/no shut destination port**, local span will stop working. As a workaround, **shut/no shut the span session** can recover it.
- SPAN source or destination is supported on any port.
- The cyclic redundancy check (CRC) is recalculated for the truncated packet.
- Tx SPAN packets are truncated to 180 Bytes (Rx SPAN mirrors the whole packets).
- The following SPAN functions are not supported:
  - IPv6 ACL filter (Tx)
  - Source VLAN Tx/Rx
  - VLAN filter Tx/Rx
  - ACL filter SPAN Tx (v4, v6)
  - CPU source (In-band SPAN)
  - Same source in multiple SPAN
  - SPAN PFC packets
  - Port-channel as destination (local or ERSPAN)
  - Source port sub-interface

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

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 3400-S Series NX-OS Interfaces Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot*/*port*
3. **switchport**
4. **switchport monitor**
5. (Optional) Repeat Steps 2 through 4 to configure monitoring on additional SPAN destinations.
6. **no monitor session** *session-number*
7. **monitor session** *session-number* [**shut**]
8. **description** *description*
9. **source** {**interface** *type* [**rx** | **tx** | **both**] | [**rx**]}
10. (Optional) **filter access-group** *acl-filter*
11. **destination interface** *type slot*/*port*
12. **no shut**
13. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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:**	Enters interface configuration mode on the selected slot and port.

	Command or Action	Purpose
	`switch(config)# interface ethernet 2/5` `switch(config-if)#`	
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* [**shut**]
	Example:
	Example:
	`switch(config)# monitor session 3 shut`
	`switch(config-monitor)#`
		Enters the monitor configuration mode. The new session configuration is added to the existing session configuration. By default, the session is created in the shut state, and the session is a local SPAN session. The optional keyword shut specifies a shut state for the selected session.

Step 8	**description** *description* **Example:** `switch(config-monitor)# description my_span_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

Step 9	**source** {**interface** *type* [**rx** | **tx** | **both**] | [**rx**]}
	Example:
	`switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx`
	Example:
	`switch(config-monitor)# source interface port-channel 2`
		You can configure one or more sources, as either a series of comma-separated entries or a range of numbers.
		You can specify the traffic direction to copy as ingress (rx), egress (tx), or both.
		For a unidirectional session, the direction of the source must match the direction specified in the session.

Step 10	(Optional) **filter access-group** *acl-filter* **Example:** `switch(config-monitor)# filter access-group ACL1`	Associates an ACL with the SPAN session.
Step 11	Required: **destination interface** *type slot*/*port* **Example:** `switch(config-monitor)# destination interface ethernet 2/5` **Example:**	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.

	Command or Action	Purpose
	`switch(config-monitor)# destination interface sup-eth 0`	
Step 12	Required: **no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 13	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-monitor)# show monitor session 3`	Displays the SPAN configuration.
Step 14	(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 (SPAN) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based SPAN. For information, see the "Configuring ACL TCAM Region Sizes" section in the *Cisco Nexus 3400-S Series NX-OS Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **udf** *udf-name offset-base offset length*
3. **hardware access-list tcam region span qualify udf** *udf-names*
4. **copy running-config startup-config**
5. **reload**
6. **ip access-list** *span-acl*
7. Enter one of the following commands:
   - **permit udf** *udf-name value mask*
   - **permit ip** *source destination* **udf** *udf-name value mask*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 span qualify udf** *udf-names* **Example:** `switch(config)# hardware access-list tcam region span qualify udf udf-x udf-y`	Attaches the UDFs to one of the following TCAM regions: • SPAN —Applies to Layer 2 & Layer 3 ports. You can attach up to 2 UDFs to a TCAM region. **Note** 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 3400-S 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**	Reloads the device.

	Command or Action	Purpose
	Example: `switch(config)# reload`	**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`
	Example:
	`switch(config-acl)# permit ip 10.0.0./24 any udf udf-x 0x02 0x0F 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.

Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Shutting Down or Resuming a SPAN Session

You can shut down SPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, SPAN sessions are created in the shut state.

You can resume (enable) SPAN sessions to resume the copying of packets from sources to destinations. In order to enable a SPAN session that is already enabled but operationally down, you must first shut it down and then enable it.

You can configure the shut and enabled SPAN session states with either a global or monitor configuration mode command.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **monitor session** {*session-range* | **all**} **shut**
3. **monitor session** *session-number*
4. [**no**] **shut**
5. (Optional) **show monitor**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.
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:

SUMMARY STEPS

1. Configure destination ports in access mode and enable SPAN monitoring.
2. Configure a SPAN session.

DETAILED STEPS

Step 1 Configure destination ports in access mode and enable SPAN monitoring.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Step 2 Configure a SPAN session.

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-monitor)# source interface port-channel 2
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Configuration Example for a Unidirectional SPAN Session

To configure a unidirectional SPAN session, follow these steps:

SUMMARY STEPS

1. Configure destination ports in access mode and enable SPAN monitoring.
2. Configure a SPAN session.

DETAILED STEPS

Step 1 Configure destination ports in access mode and enable SPAN monitoring.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Step 2 Configure a SPAN session.

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3 rx
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Configuration Example for a SPAN ACL

This example shows how to configure a SPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# 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

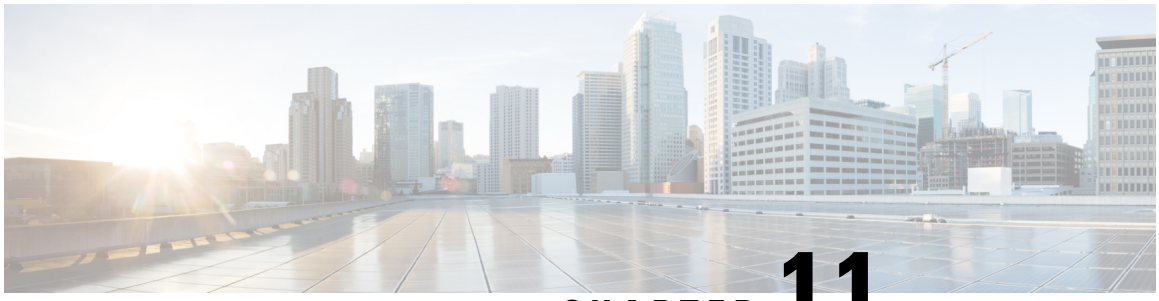
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region span 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
  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 span 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
  source interface Ethernet 1/1
  filter access-group acl-udf-pktsig
```



CHAPTER 11

Configuring ERSPAN

This chapter describes how to configure an encapsulated remote switched port analyzer (ERSPAN) to transport mirrored traffic in an IP network on Cisco NX-OS devices.

This chapter contains the following sections:

About ERSPAN

ERSPAN transports mirrored traffic over an IP network, which provides remote monitoring of multiple switches across your network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface.

ERSPAN 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
- Forward drops

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

ERSPAN Sessions

You can create ERSPAN sessions that designate sources to monitor.

Localized ERSPAN Sessions

An ERSPAN session is localized when all of the source interfaces are on the same line card.

Licensing Requirements for ERSPAN

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	ERSPAN requires no license. Any feature not included in a license package is bundled with the nx-os i is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing schen Cisco NX-OS Licensing Guide.

Prerequisites for ERSPAN

ERSPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired ERSPAN configuration. For more information, see the *Cisco Nexus 3400-S Series NX-OS Interfaces Configuration Guide*.

Guidelines and Limitations for ERSPAN

ERSPAN has the following configuration guidelines and limitations:

- ERSPAN truncation is not supported on Cisco Nexus 3400 Series switches.
- For ERSPAN session limits, see the *Cisco Nexus 3400 Series NX-OS Verified Scalability Guide*.
- Two ERSPAN destination sessions are not supported on Cisco Nexus 3400-S platform switches.
- Only ERSPAN source sessions are supported. Destination sessions are not supported.
- ERSPAN destination as a Port channel is not supported.
- 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.
- Configuring UDF based filter is supported only on Ethernet ports and Port-channels.

- 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.
- For SPAN forward drop traffic, SPAN only the packets that get dropped due to various reasons in the forwarding plane. This enhancement is supported only for ERSPAN Source session. It is not supported along with SPAN ACL and source interface.
- ERSPAN is not supported over a VXLAN overlay.
- ERSPAN works on default and nondefault VRFs.

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.

Default Settings

The following table lists the default settings for ERSPAN parameters.

Table 10: Default ERSPAN Parameters

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

Configuring ERSPAN

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.

SUMMARY STEPS

1. **configure terminal**
2. **monitor erspan origin ip-address** *ip-address* **global**
3. **no monitor session** {*session-number* | **all**}
4. **monitor session** {*session-number* | **all**} **type erspan-source** [**shut**]
5. **description** *description*

6. **source** {**interface** *type* [**tx** | **rx** |**both**] }
7. (Optional) Repeat Step 7 to configure all ERSPAN sources.
8. **destination ip** *ip-address*
9. **erspan-id** *erspan-id*
10. **vrf** *vrf-name*
11. (Optional) **ip ttl** *ttl-number*
12. (Optional) **ip dscp** *dscp-number*
13. **no shut**
14. **exit**
15. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
16. (Optional) **show running-config monitor**
17. (Optional) **show startup-config monitor**
18. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**monitor erspan origin ip-address** *ip-address* **global** **Example:** `switch(config)# monitor erspan origin ip-address 10.0.0.1 global`	Configures the ERSPAN global origin IP address.
Step 3	**no monitor session** {*session-number* | **all**} **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified ERSPAN session. The new session configuration is added to the existing session configuration.

Step 4	**monitor session** {*session-number* | **all**} **type erspan-source** [**shut**]
	Example:
	`switch(config)# monitor session 3 type erspan-source`
	`switch(config-erspan-src)#`
		Configures an ERSPAN Type II source session. By default the session is bidirectional. The optional keyword shut specifies a shut state for the selected session.

Step 5	**description** *description* **Example:** `switch(config-erspan-src)# description erspan_src_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.
Step 6	**source** {**interface** *type* [**tx** | **rx** |**both**] } **Example:** `switch(config-erspan-src)# source interface ethernet 2/1-3, ethernet 3/1 rx`	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.

	Command or Action	Purpose
	Example: `switch(config-erspan-src)# source interface port-channel 2`	For a unidirectional session, the direction of the source must match the direction specified in the session.
Step 7	(Optional) Repeat Step 7 to configure all ERSPAN sources.	—
Step 8	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1`	Configures the destination IP address in the ERSPAN session. Only one destination IP address is supported per ERSPAN source session.
Step 9	**erspan-id** *erspan-id* **Example:** `switch(config-erspan-src)# erspan-id 5`	Configures the ERSPAN ID for the ERSPAN source session. The ERSPAN range is from 1 to 1023.
Step 10	**vrf** *vrf-name* **Example:** `switch(config-erspan-src)# vrf default`	Configures the virtual routing and forwarding (VRF) instance that the ERSPAN source session uses for traffic forwarding. The VRF name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 11	(Optional) **ip ttl** *ttl-number* **Example:** `switch(config-erspan-src)# ip ttl 25`	Configures the IP time-to-live (TTL) value for the ERSPAN traffic. The range is from 1 to 255.
Step 12	(Optional) **ip dscp** *dscp-number* **Example:** `switch(config-erspan-src)# ip dscp 42`	Configures the differentiated services code point (DSCP) value of the packets in the ERSPAN traffic. The range is from 0 to 63.
Step 13	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN source session. By default, the session is created in the shut state.

Step 14	**exit**
	Example:
	`switch(config-erspan-src)# exit`
	`switch(config)#`
		Exits the monitor configuration mode.

Step 15	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config)# show monitor session 3`	Displays the ERSPAN session configuration.
Step 16	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the running ERSPAN configuration.
Step 17	(Optional) **show startup-config monitor** **Example:**	Displays the ERSPAN startup configuration.

	Command or Action	Purpose
	`switch(config)# show startup-config monitor`	
Step 18	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SPAN Forward Drop Traffic for ERSPAN Source Session

You can configure the device to match on the forwarding drop event and send the matching packets to ERSPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** {*session-number* | **all**} **type erspan-source**
3. **vrf** *vrf-name*
4. **destination ip** *ip-address*
5. **source forward-drops rx**
6. **no shut**
7. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*}

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**monitor session** {*session-number*	**all**} **type erspan-source**
		Example:
		`switch(config)# monitor session 1 type erspan-source`
		`switch(config-erspan-src)#`
			Configures an ERSPAN source session.

Step 3	**vrf** *vrf-name* **Example:** `switch(config-erspan-src)# vrf default`	Configures the VRF that the ERSPAN source session uses for traffic forwarding.
Step 4	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1`	Configures the destination IP address in the ERSPAN session. Only one destination IP address is supported per ERSPAN source session.
Step 5	**source forward-drops rx** **Example:**	Configures the SPAN forward drop traffic for the ERSPAN source session.

	Command or Action	Purpose
	`switch(config-erspan-src)# source forward-drops rx`	
Step 6	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN source session. By default, the session is created in the shut state. **Note** Only two ERSPAN source sessions can be running simultaneously.
Step 7	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} **Example:** `switch(config-erspan-src)# show monitor session 3`	Displays the ERSPAN session configuration.

Example

```
switch# config t
  switch(config)# monitor session 1 type erspan-source
  switch(config-erspan-src)# vrf default
  switch(config-erspan-src)# destination ip 40.1.1.1
  switch(config-erspan-src)# source forward-drops rx
  switch(config-erspan-src)# no shut
  switch(config-erspan-src)# show monitor session 1
```

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.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** {*session-range* | **all**} **shut**
3. **no monitor session** {*session-range* | **all**} **shut**
4. **monitor session** *session-number* **type erspan-source**
5. **shut**
6. **no shut**
7. **exit**
8. (Optional) **show monitor session all**
9. (Optional) **show running-config monitor**
10. (Optional) **show startup-config monitor**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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:** `switch(config)# show running-config monitor`	Displays the ERSPAN running configuration.
Step 10	(Optional) **show startup-config monitor** **Example:**	Displays the ERSPAN startup configuration.

	Command or Action	Purpose
	`switch(config)# show startup-config monitor`	
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 or IPv6 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.

SUMMARY STEPS

1. **configure terminal**
2. **{ ip | ipv6 } access-list** *acl-name*
3. [*sequence-number*] {**permit** | **deny**} *protocol source destination* [*protocol-value*]
4. **exit**
5. **vlan access-map** *list-name*
6. **match ip address** *acl-name*
7. **actions (drop | forward | redirect }**
8. **exit**
9. (Optional) **show ip access-lists** *name*
10. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**{ ip | ipv6 } 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* [*protocol-value*] **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.

	Command or Action	Purpose
	`switch(config-acl)# permit ip 192.168.2.0/24` `switch (config-acl)#`	The **permit** and **deny** commands support many ways of identifying traffic.

Step 4	**exit**
	Example:
	`switch (config-acl)# exit`
	`switch(config)#`
		Exits the IP ACL configuration mode and enters the global configuration mode.

Step 5	**vlan access-map** *list-name*
	Example:
	`switch(config)# permit ip 192.168.2.0/24`
	`switch(config-access-map)#`
		Creates a VLAN access map and enters the access map configuration mode.

Step 6	**match ip address** *acl-name*
	Example:
	`switch(config-access-map)# match ip address erspan-acl`
	`switch(config-access-map)#`
		Configures the access map to match IP addresses based on the IP ACL configuration.

Step 7	**actions (drop | forward | redirect }**
	Example:
	`switch(config-access-map)# action forward`
	`switch(config-access-map)#`
		Configures the access map to take action on packets whose IP address matches that of the IP ACL configuration.

Step 8	**exit**
	Example:
	`switch (config-access-map)# exit`
	`switch(config)#`
		Exits the access map configuration mode and enters the global configuration mode.

Step 9	(Optional) **show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists erpsan-acl`	Displays the ERSPAN ACL configuration.
Step 10	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config)# show monitor session 1`	Displays the ERSPAN session 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 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 packets that are defined in the criteria by the user.

Before you begin

Make sure that the appropriate TCAM region (SPAN) 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 3400-S Series NX-OS Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **udf** *udf-name offset-base offset length*
3. **hardware access-list tcam region span qualify udf** *udf-names*
4. **copy running-config startup-config**
5. **reload**
6. **ip access-list** *erspan-acl*
7. Enter one of the following commands:
   - **permit udf** *udf-name value mask*
   - **permit ip** *source destination* **udf** *udf-name value mask*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

	Command or Action	Purpose
		You can define multiple UDFs, but Cisco recommends defining only required UDFs.
Step 3	**hardware access-list tcam region span qualify udf** *udf-names* **Example:** `switch(config)# hardware access-list tcam region span qualify udf udf-x udf-y`	Attaches the UDFs to one of the following TCAM regions: • **span**—Applies to layer 2 and Layer 3 ports. You can attach up to 2 UDFs to a TCAM region. **Note** 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 3400-S 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`
	Example:
	`switch(config-acl)# permit ip 10.0.0./24 any udf udf-x 0x02 0x0F 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
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the 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
switch(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 9.1.1.2
switch(config-erspan-src)# no shut
switch(config-erspan-src)# exit
switch(config)# show monitor session 1
```

Configuration Example for an ERSPAN ACL

The examples in this sectio show how to configure ERSPAN ACLs for both IPv4 and IPv6.

This example shows how to configure an ERSPAN IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# vlan access-map erspan_filter 5
switch(config-access-map)# match ip address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# vlan access-map erspan_filter 10
switch(config-access-map)# match ip address match_12_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# filter access_group erspan_filter
```

This example shows how to configure an ERSPAN IPv6 ACL:

```
switch# configure terminal
switch(config)# ipv6 access-list match_11_pkts
switch(config-acl)# permit ipv6 permit ipv6 2040::0/32 any
```

```
switch(config-acl)# exit
switch(config)# vlan access-map erspan_filter 15
switch(config-access-map)# match ipv6 address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# filter access_group erspan_filter
```

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:

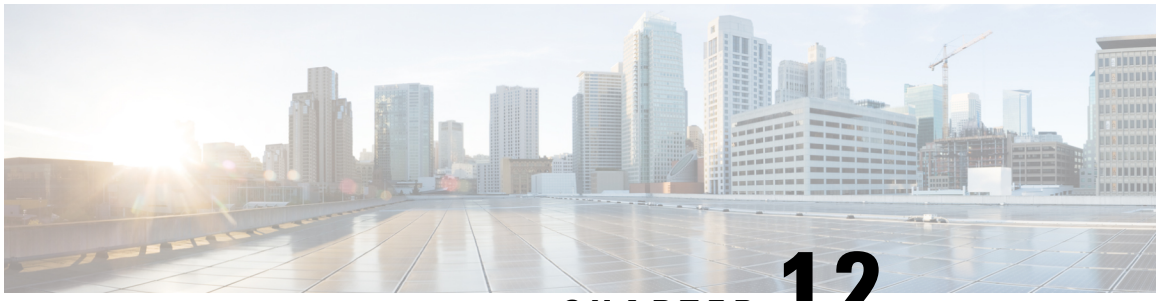
- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region span 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 span 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
```



CHAPTER 12

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 3400 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.

Virtualization Support

One instance of LLDP is supported.

Licensing Requirements for LLDP

Product	License Requirement
Cisco NX-OS	LLDP requires no license. Any feature not included in a license package is bundled with the nx-os provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing sche *Cisco NX-OS Licensing Guide*.

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 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.
- ETS Configuration and Recommendation 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
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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature lldp**
3. (Optional) **show running-config lldp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot/port*
3. [**no**] **lldp transmit**
4. [**no**] **lldp receive**
5. (Optional) **show lldp interface** *interface slot/port*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

LLDP Multi-Neighbor Support

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 on Interfaces

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.

SUMMARY STEPS

1. **configure terminal**
2. **no lldp tlv-select dcbxp**
3. [**no**] **lldp multi-neighbor**
4. **interface** *port* / *slot*
5. (Optional) [**no**] **lldp transmit**
6. (Optional) [**no**] **lldp receive**
7. (Optional) **show lldp interface**l *port* / *slot*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.
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).

SUMMARY STEPS

1. **configure terminal**
2. **no lldp tlv-select dcbxp**
3. [**no**] **lldp port-channel**
4. **interface port-channel** [*port-channel-number* | *port-channel-range*]
5. (Optional) [**no**] **lldp transmit**
6. (Optional) [**no**] **lldp receive**
7. (Optional) **show lldp interface port-channe**l *port-channel-number*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

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. **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** *port-channel-number* **Example:** `switch(config-if)# show lldp interface port-channel 3`	Displays the LLDP configuration on the port-channel.

	Command or Action	Purpose
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 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 verison, 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot*/*port*
3. **lldp dcbx version** *cee*/*ieee*/*auto*

DETAILED STEPS

	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.

	Command or Action	Purpose
		The default is set to *auto*.

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.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) [**no**] **lldp dcbx version** {**cee**| |**ieee**| |**auto**}
3. (Optional) [**no**] **lldp holdtime** *seconds*
4. (Optional) [**no**] **lldp reinit** *seconds*
5. (Optional) [**no**] **lldp timer** *seconds*
6. (Optional) **show lldp timers**
7. (Optional) [**no**] **lldp tlv-select** *tlv*
8. (Optional) **show lldp tlv-select**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	(Optional) [**no**] **lldp dcbx version** {**cee**| |**ieee**| |**auto**} **Example:** `switch(config)# lldp dcbx 3 auto`	
Step 3	(Optional) [**no**] **lldp holdtime** *seconds* **Example:** `switch(config)# lldp holdtime 200`	Specifies the amount of time in seconds that a receiving device should hold the information that is sent by your device before discarding it. The range is 10 to 255 seconds; the default is 120 seconds.
Step 4	(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 5	(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.

	Command or Action	Purpose
Step 6	(Optional) **show lldp timers** **Example:** `switch(config)# show lldp timers`	Displays the LLDP hold time, delay time, and update frequency configuration.
Step 7	(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 8	(Optional) **show lldp tlv-select** **Example:** `switch(config)# show lldp tlv-select`	Displays the LLDP TLV 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.

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.
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.

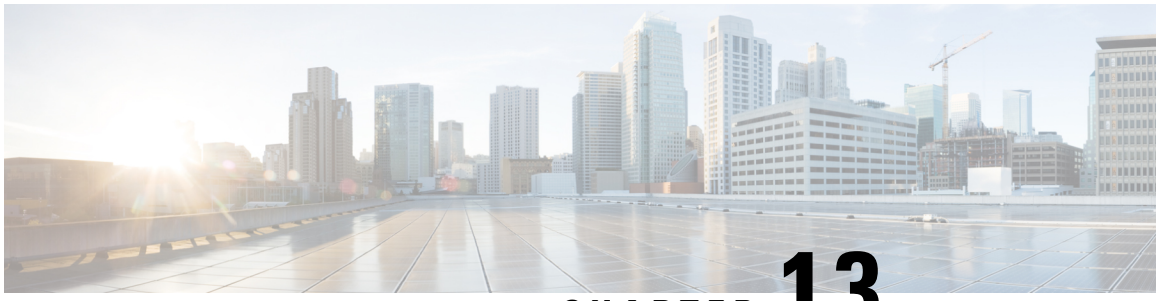
Command	Purpose
show qos dcbxp *interface slot/port*	Displays DCBXP information for a specific interface.

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 13

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.

Licensing Requirements for sFlow

Product	License Requirement
Cisco NX-OS	sFlow requires no license. Any feature not included in a license package is bundled with the nx-os ima provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, *Cisco NX-OS Licensing Guide*.

Prerequisites for sFlow

sFlow has the following prerequisites:

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

Guidelines and Limitations for sFlow

Note For scale information, see the release-specific *Cisco Nexus 3400-S NX-OS Verified Scalability Guide*.

sFlow has the following guidelines and limitations:

- 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.
- sFlow is not supported on the SVIs.
- Subinterfaces are not supported for sFlow.
- We recommend you configure the sampling rate 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 supports sampling IPv6 traffic but only on IPv4 collector ports.
- Egress sFLOW and Egress SPAN/ERSPAN cannot be enabled at the same time. The Egress sflow is disabled by default. Enabling requires a reload after configuration.

Default Settings for sFlow

The following table lists the default settings for sFlow parameters.

Table 11: 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature sflow**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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. Egress sflow is not enabled by default and the configuration has to be stored as startup configuration and system reloaded for Egress SFLOW to be enabled.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow sampling-rate** *sampling-rate*
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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:** `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 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow max-sampled-size** *sampling-size*
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**

2. [**no**] **sflow counter-poll-interval** *poll-interval*
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow max-datagram-size** *datagram-size*
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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**] **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.
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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow collector-ip** *ip-address* **vrf** *vrf* [**source** *ip-address*]
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 sampling is disabled. 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.

	Command or Action	Purpose
		• **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 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow collector-port** *collector-port*
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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**] **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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow agent-ip** *ip-address*
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 sampling is disabled on the switch. 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**	Displays the sFlow configuration.

	Command or Action	Purpose
	Example: `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 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.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **sflow data-source interface** [**ethernet** *slot*/*port*[-*port*] | **port-channel** *channel-number*]
3. (Optional) **show sflow**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

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

Verifying the sFlow Configuration

Use these commands to display the sFlow configuration.

Table 12: 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.

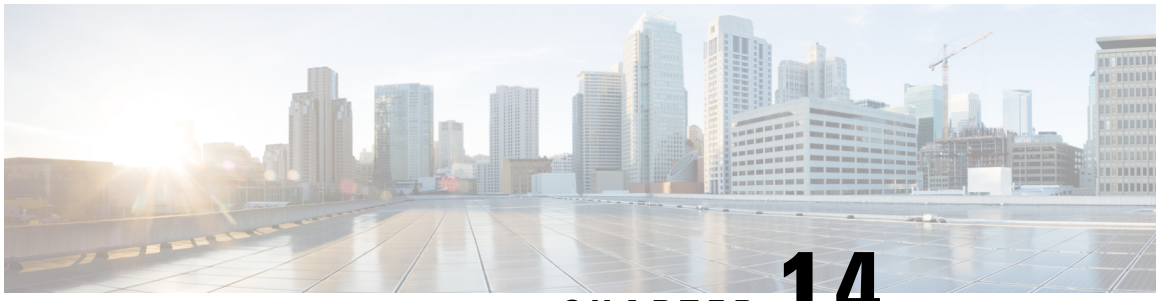
Use the following commands to clear the sFlow statistics:

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.

Additional References

Related Documents

Related Topic	Document Title
ACL TCAM regions	Configuring IP ACLs



CHAPTER 14

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 immediate issues. All defects fixed by SMUs are integrated into the maintenance releases.

For information on upgrading your device to a new feature or maintenance release, see the "Upgrading and Downgrading the Cisco Nexus 3400-S Series NX-OS Software" chapter in the *Cisco Nexus 3400-S Series NX-OS Software Upgrade and Downgrade Guide*.

For information on Cisco NX-OS optionality feature, see the "Optionality in Cisco NX-OS Software" chapter in the *Cisco Nexus 3400-S Series NX-OS Software Upgrade and Downgrade Guide*.

Note Activating an SMU does not cause any earlier SMUs, or the package to which the SMU applies, to be automatically deactivated.

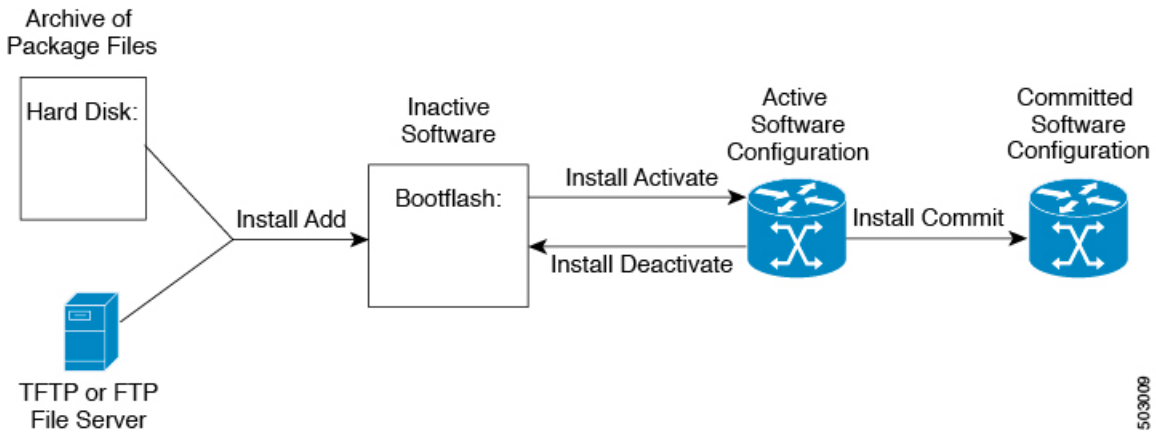
Package Management

The general procedure for adding and activating SMU packages on the device is as follows:

1. Copy the package file or files to a local storage device or file server.
2. Add the package or packages on the device using the **install add** command.
3. Activate the package or packages on the device using the **install activate** command.
4. Commit the current set of packages using the **install commit** command.
5. (Optional) Deactivate and remove the package.

The following figure illustrates the key steps in the package management process.

Figure 5: Process to Add, Activate, and Commit SMU Packages



Impact of Package Activation and Deactivation

The activation or deactivation of an SMU package can have an immediate impact on the system. The system can be affected in the following ways:

- New processes might be started.
- Running processes might be stopped or restarted.
- All processes in the line cards might be restarted. Restarting processes in the line cards is equivalent to a soft reset.
- The line cards might reload.
- No processes in the line cards might be affected.

Note You must address any issues that result from the revised configuration and reapply the configuration, if necessary.

Tip After the activation process completes, enter the **show install log** command to display the process results.

Prerequisites for SMUs

These prerequisites must be met for a package to be activated or deactivated:

- You must be in a user group associated with a task group that includes the proper task IDs. If you suspect a user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Verify that all line cards are installed and operating properly. For example, do not activate or deactivate packages while line cards are booting, while line cards are being upgraded or replaced, or when you anticipate an automatic switchover activity.

Guidelines and Limitations for SMUs

SMUs have the following guidelines and limitations:

- Some packages require the activation or deactivation of other packages. If the SMUs have dependencies on each other, you cannot activate them without first activating the previous ones.
- The package being activated must be compatible with the current active software set.
- Activation is performed only after the package compatibility checks have been passed. If a conflict is found, an error message displays.
- You can activate or deactivate multiple SMUs with a tarball SMU.
- While a software package is being activated, other requests are not allowed to run on any of the impacted nodes. Package activation is completed when a message similar to this one appears:

  ```
  Install operation 1 completed successfully at Thu Jan 9 01:19:24 2014
  ```

- Each CLI install request is assigned a request ID, which can be used later to review the events.
- If you perform a software maintenance upgrade and later upgrade your device to a new Cisco NX-OS software release, the new image will overwrite both the previous Cisco NX-OS release and the SMU package file.
- The SMU package file is named `nxos.CSCab00001-n9k_ALL-1.0.0-7.0.3.I5.1.lib32_n9000.rpm`, which support both n9k_EOR and n9k_TOR platforms.

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.

SUMMARY STEPS

1. **show logging logfile | grep -i "System ready"**
2. **show install active**
3. **show module**
4. **show clock**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show logging logfile | grep -i "System ready"** **Example:** `switch# show logging logfile | grep -i "System ready"`	Displays if your system is up. Use this command to verify that the system is ready for SMU package installation. Configuring install commands before the system is ready, may result with an "Install operation 11 failed because cannot lock config" error message.
Step 2	**show install active** **Example:** `switch# show install active`	Displays the active software on the device. Use this command to determine what software should be added on the device and to compare to the active software report after installation operations are complete.
Step 3	**show module** **Example:** `switch# show module`	Confirms that all modules are in the stable state.
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:

SUMMARY STEPS

1. Log in to Cisco.com.
2. Go to the Download Software page at this URL: http://software.cisco.com/download/navigator.html
3. In the Select a Product list, choose **Switches** > **Data Center Switches** > **Cisco Nexus 9000 Series Switches** > *model*.
4. Choose the appropriate SMU file for your device and click **Download**.

DETAILED STEPS

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 "Configuring SSH and Telnet" chapter in the *Cisco Nexus 3400-S 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 13: 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.

Command	Purpose
copy ftp://*username*:*password*@*hostname-or-ipaddress*/*directory-path*/*filename* **bootflash:** `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:`	Copies the package file from the FTP server to the bootflash:. • *username*—The username of the user who has access privileges to the directory in which the package file is stored. • *password*—The password associated with the username of the user who has access privileges to the directory in which the package file is stored. If a password is not provided, the networking device accepts anonymous FTP. • *hostname-or-ipaddress*—The hostname or IP address of the network file server. • *directory-path*—The network file server path that leads to the package file to be added. The specified directory should be a directory under the home directory of the user. In this example, the file being downloaded is in a subdirectory called "images" in the home directory of the user "john." **Note** For FTP services, *directory-path* is the directory relative to the *username* home directory. If you want to specify an absolute path for the directory, you must add a “/” following the server address. • *filename*—The name of the package file that you want to add.

Command	Purpose
copy sftp://*hostname-or-ipaddress*/*directory-path*/*filename* **bootflash:** `switch# copy sftp://10.1.1.1/images/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 199.

SUMMARY STEPS

1. Connect to the console port and log in.
2. (Optional) **dir bootflash:**
3. **install add** *filename* [**activate**]
4. (Optional) **show install inactive**
5. **install activate** *filename*
6. Repeat Step 5 until all packages are activated.
7. (Optional) **show install active**

DETAILED STEPS

	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.

	Command or Action	Purpose
		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.
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 202.

SUMMARY STEPS

1. **install commit** *filename*
2. (Optional) **show install committed**

DETAILED STEPS

	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.

SUMMARY STEPS

1. Connect to the console port and log in.
2. **install deactivate** *filename*
3. (Optional) **show install inactive**
4. (Optional) **install commit**
5. (Optional) **install remove** {*filename* | **inactive**}

DETAILED STEPS

	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.
Step 3	(Optional) **show install inactive** **Example:** `switch# show install inactive`	Displays the inactive packages on the device.

	Command or Action	Purpose
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 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`
		Removes the inactive package.
		• Only inactive packages can be removed.
		• Packages can be removed only if they are deactivated from all line cards in the device.
		• The package deactivation must be committed.
		• To remove a specific inactive package from a storage device, use the **install remove** command with the *filename* argument.
		• To remove all inactive packages from all nodes in the system, use the **install 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.

SUMMARY STEPS

1. (Optional) **show install packages**
2. **run bash**
3. **cd /rpms**
4. **ls ****feature****
5. **cp** *filename* **/bootflash**
6. **exit**
7. **install add bootflash:***filename* **activate downgrade**
8. (Optional) **show install packages | i** *feature*

DETAILED STEPS

	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.

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`
		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.

	Command or Action	Purpose
	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	
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.

SUMMARY STEPS

1. Download the SMU package file for Guest Shell Bash from Cisco.com.
2. Copy the SMU package file to the bootflash: of the switch.
3. **guestshell**
4. **sudo rpm -Uvh** /bootflash/*filename*
5. **rpm -qa** | **grep bash**
6. **guestshell sync**

DETAILED STEPS

	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 199.
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 199.

Step 3	**guestshell**
	Example:
	`switch# guestshell`
	`guestshell:~$`
		Accesses the Guest Shell.

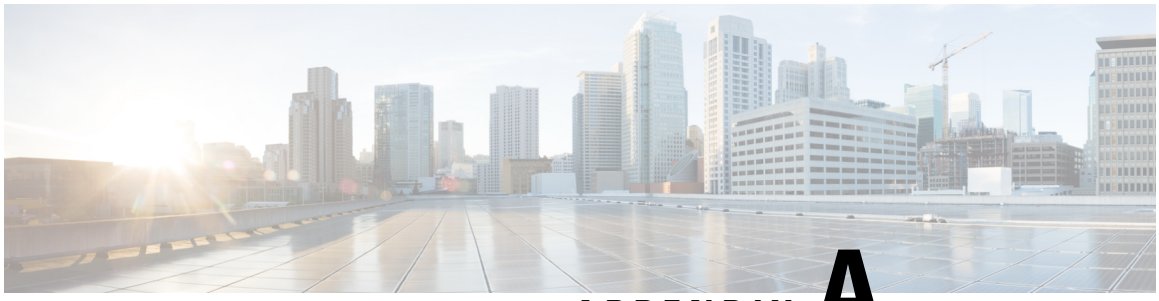
Step 4	**sudo rpm -Uvh** /bootflash/*filename*
	Example:
	`guestshell:~$ sudo rpm -Uvh /bootflash/bash-4.2-r8.x86_64.rpm`
	`Preparing...`
	`########################################## [100%]`
	`1:bash`
		Upgrades the existing Bash file in the Guest Shell.

	Command or Action	Purpose
| | ```
########################################## [100%]
update-alternatives: Linking //bin/sh to /bin/bash
``` | |
| **Step 5** | **rpm -qa \| grep bash**<br>**Example:**<br>```
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**<br>**Example:**<br>```
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 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+'
``` | 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.<br>**Note** The new Bash file is preserved after a Guest Shell reboot or Guest Shell disable+enable. However, you need to reinstall the Guest Shell Bash SMU package file after a Guest Shell destroy+enable. |

Additional References

Related Documents

Related Topic	Document Title
Software upgrades	*Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*
```



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.

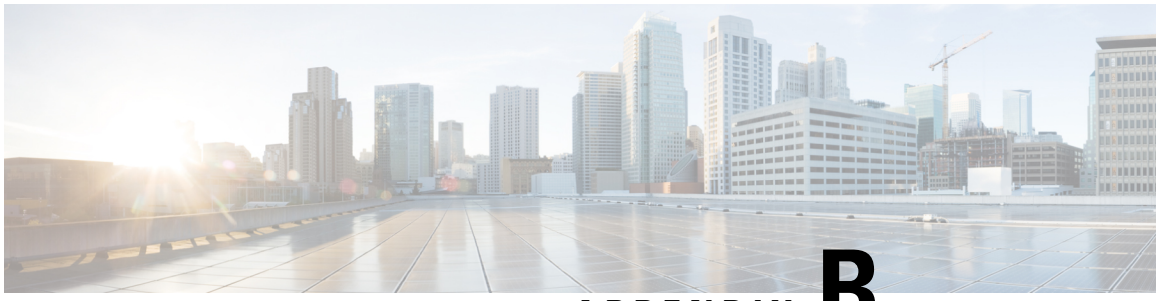
This appendix includes the following sections:

IETF RFCs Supported by Cisco NX-OS System Management

This appendix lists the IETF RFCs for system management supported in Cisco NX-OS.

RFCs	Title
RFC 2819	*Remote Network Monitoring Management Information Ba*
RFC 3411 and RFC 3418	*An Architecture for Describing Simple Network Managem (SNMP) Management Frameworks*

IETF RFCs Supported by Cisco NX-OS System Management



APPENDIX B

Embedded Event Manager System Events and Configuration Examples

This appendix describes the Embedded Event Manager (EEM) system policies, events, and policy configuration examples.

This appendix includes the following sections:

EEM System Policies

The following table lists the Embedded Event Manager (EEM) system policies.

Event	Description
__BootupPortLoopback	Do CallHome, Error-disable affected ports, log error testing on affected ports after 1 consecutive failures of GOLD "BootupPortLoopback" test
__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

Embedded Event Manager System Events and Configuration Examples

Event	Description
__asicmem	Do CallHome and log error when GOLD "AsicMemory" test fails. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
__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
__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
__fpgareg	Do CallHome, log error, disable further HM testing after 20 consecutive failures of GOLD "FpgaRegTest" test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.

Event	Description
__L2ACLRedirect	Do CallHome, log error, disable further HM testing after 10 consecutive failures of L2ACLRedirect test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
__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
__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

Event	Description
__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.
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.

EEM Event	Description
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:

SUMMARY STEPS

1. Create an object to track the status of Ethernet interface 3/23.
2. Configure an EEM event to shut Ethernet interface 1/2 when the tracking object shuts down.
3. Configure an EEM event to bring up Ethernet interface 1/2 when Ethernet interface 3/23 comes up.

DETAILED STEPS

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 2.0 syslog msg severity alert "##### 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