Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x

March 31, 2011

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

Text Part Number: OL-20086-01

TTHE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, CCSI, Cisco Eos, Cisco Explorer, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Nurse Connect, Cisco Pulse, Cisco SensorBase, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco TrustSec, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flipshare (Design), Flip Ultra, Flip Video, Flip Video (Design), Instant Broadband, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Capital, Cisco Capital (Design), Cisco:Financed (Stylized), Cisco Store, Flip Gift Card, and One Million Acts of Green are service marks; and Access Registrar, Aironet, AllTouch, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Lumin, Cisco Nexus, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Continuum, EtherFast, EtherSwitch, Event Center, Explorer, Follow Me Browsing, GainMaker, iLYNX, IOS, iPhone, IronPort, the IronPort logo, Laser Link, LightStream, Linksys, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, PCNow, PIX, PowerKEY, PowerPanels, PowerTV, PowerTV (Design), PowerVu, Prisma, ProConnect, ROSA, SenderBase, SMARTnet, Spectrum Expert, StackWise, WebEx, and the WebEx logo are registered trademarks of Cisco and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1002R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x

CONTENTS

Send document comments to nexus7k-docfeedback@cisco.com.

Send document comments to nexus7k-docfeedback@cisco.com.

Send document comments to nexus7k-docfeedback@cisco.com.

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x*. The latest version of this document is available at the following Cisco website: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_2/nx-os/system_management/configuration/guide/sm_nx_os_cli.html

To check for additional information about Cisco NX-OS Release 4.2, see the *Cisco NX-OS Release Notes* available at the following Cisco website: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_2/nx-os/release/notes/42_nx-os_release_note.html

Table 1 summarizes the new and changed features for the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x*, and tells you where they are documented.

Table 1 New and Changed Features for Release 4.2

Feature	Description	Changed in Release	Where Documented
Memory thresholds	Changed the minor, severe, and critical default memory thresholds from 70, 80, and 90 to 85, 90, and 95.	4.2(4)	Chapter 13, "Configuring Memory Thresholds"
CDP support for VTP domain name	CDP advertises the VLAN Trunking Protocol (VTP) type-length-value field (TLV) in CDP version-2 packets.	4.2(1)	Chapter 4, "VTP Feature Support"
CFS Distribution for NTP	You can use Cisco Fabric Services (CFS) to distribute an NTP configuration to all CFS-enabled switches in the network.	4.2(1)	Chapter 3, "Distributing NTP Using CFS" Chapter 2, "Enabling CFS to Distribute NTP Configurations"
Automatically generated system checkpoints	The software automatically generates a system checkpoint when disabling a feature or Layer 3 protocol, or license expiration could cause loss of configuration information.	4.2(1)	Chapter 7, "Automatically Generated System Checkpoints"
Checkpoints and rollback	The checkpoint and rollback features now provide full support for high availability.	4.2(1)	Chapter 7, "High Availability."
Filter SNMP requests by community using an ACL	You can assign an ACL to an SNMP community to filter SNMP requests	4.2(1)	Chapter 10, "Filtering SNMP Requests"

Table 1 New and Changed Features for Release 4.2 (continued)

Feature	Description	Changed in Release	Where Documented
IPv6 support	Supports configuring IPv6 SNMP hosts.	4.2(1)	Chapter 10, "Configuring SNMP."
SNMP notifications added	The **snmp-server enable traps** command is added**.** The **snmp-server enable traps** *trap-arg-scope-global* command is added.	4.2(1)	Chapter 10, "Enabling SNMP Notifications"
Use interfaces for SNMP notification receivers	You can use the **snmp-server host source-interface** command to assign a host on an interface**.** You can use the **snmp-server source-interface** command to designate an interface to receive SNMP notifications.	4.2(1)	Chapter 10, "Configuring SNMP Notification Receivers"
Diagnostics PortLoopback and StandbyFabricLoopback	The generic online diagnostics (GOLD) now include PortLoopback (test ID 5) and StandbyFabricLoopback (test ID 16).	4.2(1)	Chapter 12, "Runtime or Health Monitoring Diagnostics"
EEM events	You can use the **event module status** command to trigger an event when module status varies between online and offline. You can use the **event sysmgr memory** command to trigger an event. You can use the **event sysmgr switchover** command to trigger an event.	4.2(1)	Chapter 13, "Configuring Event Statements"
Layer 2 NetFlow	You can define Layer 2 keys in flexible NetFlow records that you can use to capture flows in Layer 2 interfaces.	4.2(1)	Chapter 16, "Guidelines and Limitations" Chapter 16, "Configuring Layer 2 NetFlow"
CFS	Corrected examples in TACACS+ configuration.	4.1(4)	Chapter 2, "Enabling CFS to Distribute TACACS+ Configurations"
Rollback	Added the following note: **Note** If you make a configuration change during atomic rollback, the rollback will fail.	4.1(4)	Chapter 7, "Implementing a Rollback"
Call Home	Added new commands to support HTTP and HTTPS: **destination profile http** and **destination profile transport-method.** **Note** These command are not distributable with CFS. As a workaround, enter these commands after the **commit** command.	4.1(3)	Chapter 6, "Modifying a Destination Profile"
Call Home	Updated the Prerequisites and Guidelines and Limitations sections.	4.1(3)	Chapter 6, "Prerequisites for Call Home" Chapter 6, "Guidelines and Limitations"

Table 1 New and Changed Features for Release 4.2 (continued)

Feature	Description	Changed in Release	Where Documented
Rollback	Rollback fails for NetFlow if during rollback, you try to modify a record that is programmed in the hardware. Rollback is not supported for checkpoints across software versions.	4.1(3)	Chapter 7, "Guidelines and Limitations"
Rollback	The default rollback type is atomic.	4.1(3)	Chapter 7, "Rollback Overview" Chapter 7, "Implementing a Rollback"
EEM	Added a configuration section for memory thresholds. Added a table of system policies.	4.1(3)	Chapter 13, "Configuring Memory Thresholds" "Embedded Event Manager System Events and Configuration Examples"
SPAN	Added a table of SPAN session limits.	4.1(3)	Chapter 14, "Guidelines and Limitations"
NetFlow	Rollback fails for NetFlow if during rollback, you try to modify a record that is programmed in the hardware.	4.1(3)	Chapter 16, "Guidelines and Limitations"
CFS	Cisco Fabric Services (CFS) distributes data, including configuration changes, to all Cisco NX-OS devices in a network.	4.1(2)	Chapter 2, "Configuring CFS Distribution"
SPAN	Cisco Ethernet switched port analyzer (SPAN) destinations are enhanced to support intrusion detection by allowing the following: • Injecting packets to disrupt a TCP packet stream. • Enabling a forwarding engine to learn the MAC address of the IDS.	4.1(2)	Chapter 14, "Configuring SPAN"

Preface

This preface describes the audience, organization and conventions of the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x*. It also provides information on how to obtain related documentation.

This chapter includes the following sections:

- Audience, page xix
- Document Organization, page xix
- Document Conventions, page xx
- Related Documentation, page xxi
- Obtaining Documentation and Submitting a Service Request, page xxii

Audience

This publication is for experienced network administrators who configure and maintain Cisco NX-OS devices.

Document Organization

This document is organized into the following chapters:

Title	Description
Chapter 1, "Overview"	Provides an overview of the features in this document.
Chapter 2, "Configuring CFS"	Describes how to use Cisco Fabric Services (CFS) to distribute data, including configuration changes, to all Cisco NX-OS devices in a network.
Chapter 3, "Configuring NTP"	Describes how to configure the Network Time Protocol (NTP).
Chapter 4, "Configuring CDP"	Describes how to configure the Cisco Discovery Protocol (CDP).
Chapter 5, "Configuring System Message Logging"	Describes how to configure logging for system messages.

Send document comments to nexus7k-docfeedback@cisco.com.

Title	Description
Chapter 6, "Configuring Smart Call Home"	Describes how to configure the smart Call Home feature for e-mail-based notification of critical system policies.
Chapter 7, "Configuring Rollback"	Describes how to create configuration snapshots with the rollback feature and how to apply commands in batch mode with the Session Manager.
Chapter 8, "Configuring Session Manager"	Describes how to apply commands in batch mode with the Session Manager.
Chapter 9, "Configuring the Scheduler"	Describes how to schedule batch configuration jobs.
Chapter 10, "Configuring SNMP"	Describes how to configure SNMP and enable SNMP notifications.
Chapter 11, "Configuring RMON"	Describes how to monitor the device by configuring RMON alarms and events.
Chapter 12, "Configuring Online Diagnostics"	Describes how to configure online diagnostics to monitor the software and hardware.
Chapter 13, "Configuring the Embedded Event Manager"	Describes how to configure the Embedded Event Manager.
Chapter 15, "Configuring Onboard Failure Logging"	Describes how to configure on-board failure logging to log failure data to persistent storage.
Chapter 14, "Configuring SPAN"	Describes how to configure SPAN to monitor traffic into and out of a port.
Chapter 16, "Configuring NetFlow"	Describes how to configure NetFlow to gather statistics on input and output traffic.
Appendix A, "IETF RFCs supported by Cisco NX-OS System Management"	Lists supported IETF RFCs.
Appendix B, "Embedded Event Manager System Events and Configuration Examples"	Lists the EEM system policies.

Document Conventions

Command descriptions use these conventions:

Convention	Description
boldface font	Commands and keywords are in boldface.
italic font	Arguments for which you supply values are in italics.
[ ]	Elements in square brackets are optional.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Screen examples use these conventions:

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

This document uses the following conventions:

Note Means reader *take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation

Cisco NX-OS includes the following documents:

Release Notes

Cisco Nexus 7000 Series NX-OS Release Notes, Release 4.2

NX-OS Configuration Guides

Cisco Nexus 7000 Series NX-OS Getting Started with Virtual Device Contexts

Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x

Cisco Nexus 7000 Series NX-OS Software Upgrade and Downgrade Guide, Release 4.x

Cisco NX-OS Licensing Guide

Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide

Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x

Cisco NX-OS XML Management Interface User Guide, Release 4.x

Cisco NX-OS System Messages Reference

Cisco Nexus 7000 Series NX-OS MIB Quick Reference

NX-OS Command References

Cisco Nexus 7000 Series NX-OS Command Reference Master Index

Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference

Cisco Nexus 7000 Series NX-OS Interfaces Command Reference

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference

Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference

Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference

Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference

Cisco Nexus 7000 Series NX-OS Security Command Reference

Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference

Cisco Nexus 7000 Series NX-OS System Management Command Reference

Other Software Document

Cisco Nexus 7000 Series NX-OS Troubleshooting Guide

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

CHAPTER 1

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:

Cisco NX-OS Device Configuration Methods

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

Figure 1-1 shows the device configuration methods available to a network user.

Figure 1-1 Cisco NX-OS Device Configuration Methods

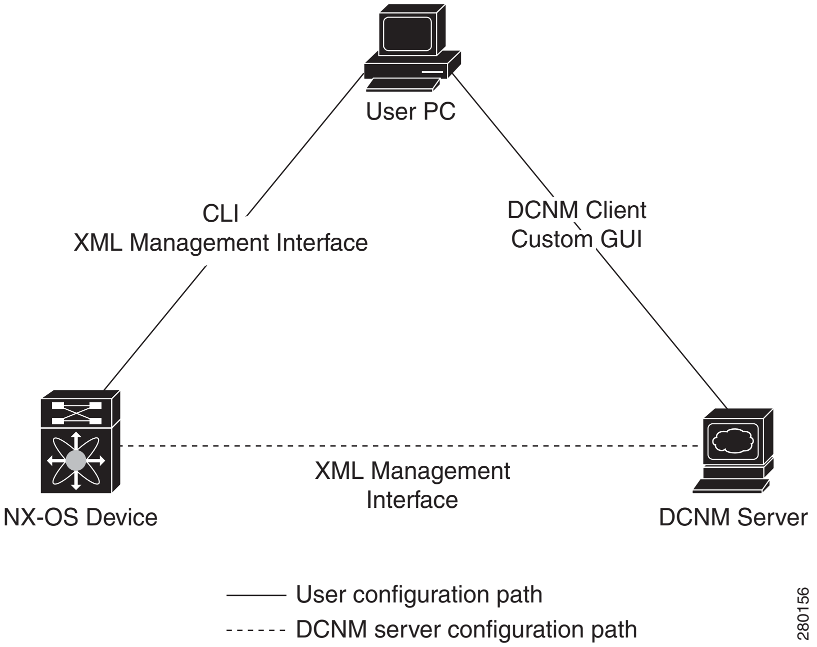


Table 1-1 lists the configuration method and the document where you can find more information.

Table 1-1 Configuration Methods Book Links

Configuration Method	Document
CLI from SSH[1], Telnet session or console port	*Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.x*
XML management interface	*Cisco NX-OS XML Management Interface User Guide, Release 4.x*
DCNM client	*Cisco DCNM Fundamentals Configuration Guide, Release 4.2*
User-defined GUI	*Cisco DCNM Web Services API Guide, Release 4.2*

[1] Secure shell (SSH).

This section includes the following topics:

- Configuring with CLI or XML Management Interface, page 1-3
- Configuring with DCNM or a Custom GUI, page 1-3

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 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.x*.
- 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 DCNM or a Custom GUI

You can configure Cisco NX-OS devices using the DCNM client or from your own GUI as follows:

- DCNM Client—You can configure devices using the DCNM client, which runs on your local PC and uses web services on the DCNM server. The DCNM server configures the device over the XML management interface. For more information about the DCNM client, see the *Cisco DCNM Fundamentals Configuration Guide*.
- Custom GUI—You can create your own GUI to configure devices using the DCNM web services application program interface (API) on the DCNM server. You use the SOAP protocol to exchange XML-based configuration messages with the DCNM server. The DCNM server configures the device over the XML management interface. For more information about creating custom GUIs, see the *Cisco DCNM Web Services API Guide, Release 4.2*.

Cisco Fabric Services

Cisco Fabric Services (CFS) is a Cisco proprietary feature that distributes data, including configuration changes, to all Cisco NX-OS devices in a network. For more information about CFS, see Chapter 2, “Configuring CFS.”

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. For more information about NTP, see Chapter 3, “Configuring NTP.”

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. For more information about CDP, see Chapter 4, "Configuring CDP."

System Messages

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

System message logging is based on RFC 3164. For more information about the system message format and the messages that the device generates, see the *Cisco NX-OS System Messages Reference*.

For information about configuring system messages, see Chapter 5, "Configuring System Message Logging."

Call Home

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

For information about configuring Call Home, see Chapter 6, "Configuring Smart Call Home."

Rollback

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

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

For more information, see the Chapter 7, "Configuring Rollback."

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.

For more information, see the Chapter 8, "Configuring Session Manager."

Scheduler

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

For more information, see Chapter 9, "Configuring the Scheduler."

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.

For more information, see Chapter 10, "Configuring SNMP."

RMON

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

For more information, see Chapter 11, "Configuring RMON."

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.

For information about configuring online diagnostics, see Chapter 12, "Configuring Online Diagnostics."

Embedded Event Manager

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

For information about configuring EEM, see Chapter 13, "Configuring the Embedded Event Manager."

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.

For information about configuring SPAN, see Chapter 14, "Configuring SPAN."

On-Board 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. For information about configuring OBFL, see Chapter 15, "Configuring Onboard Failure Logging."

NetFlow

NetFlow allows you to identify packet flows for both ingress and egress IP packets and provide statistics based on these packet flows. NetFlow does not require any change to either the packets themselves or to any networking device.

For information about configuring NetFlow, see Chapter 16, "Configuring NetFlow."

Troubleshooting Features

Cisco NX-OS provides troubleshooting tools such as ping, traceroute, Ethanalyzer, and the Blue Beacon feature. See the *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide* for details on these features.

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 a file transfer utility such as Trivial File Transfer Protocol (TFTP) by entering the **system cores** command.
- CISCO-SYSTEM-MIB contains a table for cores (cseSwCoresTable).

For information on collecting and using the generated information relating to service failures, see the *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide*.

CHAPTER 2

Configuring CFS

This chapter describes how to use Cisco Fabric Services (CFS), a Cisco proprietary feature that distributes data, including configuration changes, to all Cisco NX-OS devices in a network.

This chapter includes the following sections:

Information About CFS

You can use CFS over IP (CFSoIP) to distribute and synchronize a configuration on one Cisco device or with all other Cisco devices in your network. CFSoIP provides you with consistent and, in most cases, identical configurations and behavior in your network.

This section includes the following topics:

Merging Application Databases

When a new device is detected in your network, CFS manages the merging, or synchronizing, of its configuration with that of the other devices. CFS also coordinates and minimizes the number of merges by designating one device to manage merges per application per region. The other devices do not play any role in the merge process.

During a merger of two networks, their designated managers exchange configuration databases. The application on one of them merges the databases, decides if the merger is successful, and notifies all other devices.

If the merger is successful, the merged database is distributed to all devices in the combined fabric and the entire new fabric emerges in a consistent state. You can recover from a merge failure by starting a distribution from any device in the new fabric. This distribution restores all peers in the fabric to the same configuration database.

Applications that Use CFS to Distribute Configuration Changes

CFS distributes configuration changes for the applications shown in Table 2-2.

Table 2-1 CFS-Supported Applications

Application	Default state
RADIUS	Disabled
TACACS+	Disabled
User and administrator roles	Disabled
Call Home	Disabled
NTP	Disabled

CFS Distribution

CFS distributes configuration changes to multiple devices in a defined region or across a complete network.

The following steps provide an overview of how CFS distributes application configurations.

1. You enable CFS to distribute configurations for an application, such as Call Home.
2. You enter a command to change the configuration for a CFS application, such as Call Home.
3. CFS checks if an active fabric lock indicates that a configuration change is already in progress for this application.

Note Only one CFS session for an application can be active at a time. CFS uses locks to enforce this restriction. Distribution is not allowed to start if locks are in place for the application anywhere else in the fabric.

4. One of the following occurs:
   - If an active fabric lock exists for this application, CFS rejects the command. No changes are permitted until the existing fabric lock is released.
   - If there is not an active fabric lock for this application, then CFS starts a session and locks the fabric for this application.
5. You enter the remaining configuration commands for the application.
6. You commit the configuration by using the **commit** command.
7. CFS distributes the configuration and releases the lock.

CFS Regions

A CFS region is a user-defined subset of devices for a given feature or application. You will usually define regions to localize or restrict distribution based on devices that are close to one another.

When a network covers many geographies with many different administrators who are responsible for subsets of devices, you can manage the physical scope of an application by setting up a CFS region.

CFS regions are identified by numbers 0 through 200. Region 0 is the default region. You can configure region number 1 through 200.

Note If a feature is moved, that is, assigned to a new region, its scope is restricted to that region and it ignores all other regions for distribution or merging purposes.

You can set up a CFS region to distribute configurations for multiple features. However, on a given device, you can configure only one CFS region at a time to distribute the configuration for a given feature. Once you assign a feature to a CFS region, its configuration cannot be distributed within another CFS region.

Note The default region is used to distribute changes to all devices in a fabric. Region 0 is reserved as the default region and contains every device in the fabric. If you remove an application from a region and do not assign it to a different region, it is added to the default region (region 0).

High Availability

Stateless restarts are supported for CFS. After a reboot or a supervisor switchover, the running configuration is applied. For more information on high availability, see the *Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide*.

Virtualization Support

CFS is configured per VDC.

When you access Cisco NX-OS, it places you in the default VDC unless you specify a different VDC. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for CFS

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

Prerequisites for CFS

CFS has the following prerequisites:

- CFS is enabled by default. All devices in the fabric must have CFS enabled or they do not receive distributions.
- If CFS is disabled for an application, then that application does not distribute any configuration and it does not accept a distribution from other devices in the fabric.

Guidelines and Limitations

CFS has the following configuration guidelines and limitations:

- If the virtual port channel (vPC) feature is enabled for your device, do not disable CFS over Ethernet.

⚠ **Caution** CFS over Ethernet must be enabled for the vPC feature to work.

- CFS distributions for application data use directed unicast.
- All CFS over IP enabled devices with similar multicast addresses form one CFS over IP fabric.
- Make sure that CFS is enabled for the applications you want to configure. For detailed information, see the "Enabling CFS Distribution for Applications" procedure on page 2-5.
- Any time you lock a fabric, your username is remembered across restarts and switchovers.
- Any time you lock a fabric, configuration changes attempted by anyone else are rejected.
- While a fabric is locked, the application holds a working copy of configuration changes in a pending database or temporary storage area—not in the running configuration.
- Configuration changes that have not been committed yet (still saved as a working copy) are not in the running configuration and do not display in the output of **show** commands.
- The working copy overwrites the running configuration when you commit the changes.
- If you start a CFS session that requires a fabric lock but forget to end the session, an administrator can clear the session. For more information, see the "Clearing a Locked Session" procedure on page 2-19.
- CFSoIP and CFSoE are not supported for use together.
- CFS regions can be applied only to CFSoIP and CFSoFC clients.

- An empty commit is allowed if configuration changes are not previously made. In this case, the **commit** command results in a session that acquires locks and distributes the current database.
- You can only use the **commit** command on the specific device where the fabric lock was acquired.

Configuring CFS Distribution

This section describes how to configure CFS and includes the following topics:

- Enabling CFS Distribution for Applications, page 2-5
- Specifying a CFS Distribution Mode, page 2-10
- Configuring an IP Multicast Address for CFS Over IP, page 2-11
- Configuring CFS Regions, page 2-12
- Creating and Distributing a CFS Configuration, page 2-18
- Discarding a Configuration, page 2-20
- Disabling CFS Distribution Globally, page 2-21

Enabling CFS Distribution for Applications

This section includes the following topics:

- Enabling CFS to Distribute Call Home Configurations, page 2-5
- Enabling CFS to Distribute RADIUS Configurations, page 2-6
- Enabling CFS to Distribute TACACS+ Configurations, page 2-7
- Enabling CFS to Distribute Role Configurations, page 2-8
- Enabling CFS to Distribute NTP Configurations, page 2-9

Note See the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x* for more information on CFS for RADIUS, TACACS+, and roles. See Chapter 6, "Configuring Smart Call Home" for more information on Call Home, and see Chapter 3, "Configuring NTP" for more information on NTP.

Enabling CFS to Distribute Call Home Configurations

You can enable CFS to distribute Call Home configurations.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **distribute**

4. **show** *application_name* **status**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch# config t`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	`switch(config)#` **`callhome`**
	
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Places you in callhome configuration mode.

Step 3	`switch(config)#` **`distribute`**
	
	Example:
	`switch(config-callhome)# distribute`
	`switch(config-callhome)#`
		Enables CFS to distribute Call Home configuration updates.

Step 4	**`show`** *`application_name`* **`status`**
	
	Example:
	`switch(config-callhome)# show callhome status`
		(Optional) For the specified application, displays the CFS distribution status.

Step 5	**`copy running-config startup-config`**
	
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable CFS to distribute Call Home configurations:

```
switch(config)# callhome
switch(config-callhome)# distribute
switch(config-callhome)# show callhome status
Distribution : Enabled
switch(config-callhome)# copy running-config startup-config
[########################################] 100%
```

Enabling CFS to Distribute RADIUS Configurations

You can enable CFS to distribute RADIUS configurations.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **radius distribute**

3. **show radius status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Places you in global configuration mode.
Step 2	switch(config)# **radius distribute** **Example:** switch(config)# radius distribute	For the specified application, enables the device to receive configuration updates that are distributed through CFS.
Step 3	**show radius status** **Example:** switch(config)# show radius status	(Optional) For the specified application, displays the CFS distribution status.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable CFS to distribute RADIUS configurations:

```
switch(config)# radius distribute
switch(config)# show radius status
Distribution : Enabled
switch(config)# copy running-config startup-config
[########################################] 100%
```

Enabling CFS to Distribute TACACS+ Configurations

You can enable CFS to distribute TACACS+ configurations.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **tacacs+ distribute**
3. **show tacacs+ status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	`switch(config)#` **`tacacs+ distribute`**
	Example:
	`switch(config)# tacacs+ distribute`
		Enables CFS to distribute configuration updates for TACACS+.

Step 3	**`show tacacs+ status`**
	Example:
	`switch(config)# show tacacs+ status`
		(Optional) Displays the CFS distribution status for TACACS+.

Step 4	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable CFS to distribute TACACS+ configurations:

```
switch(config)# tacacs+ distribute
switch(config)# show tacacs+ status
Distribution : Enabled
Last operational state: No session
switch(config)# copy running-config startup-config
[########################################] 100%
```

Enabling CFS to Distribute Role Configurations

You can enable CFS to distribute role configurations.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **role distribute**
3. **show role status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	`switch(config)#` **`role distribute`**
	Example:
	`switch(config)# role distribute`
		Enables CFS to distribute role configurations.

Step 3	**`show role status`**
	Example:
	`switch(config)# show role status`
		(Optional) Displays the CFS distribution status.

Step 4	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable CFS to distribute Call Home configurations:

```
switch(config)# role distribute
switch(config)# show role status
Distribution : Enabled
switch(config)# copy running-config startup-config
[########################################] 100%
```

Enabling CFS to Distribute NTP Configurations

You can enable CFS to distribute NTP configurations.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **ntp distribute**
3. **show** *application_name* **status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**ntp distribute** **Example:** `switch(config)# ntp distribute`	Enables CFS to distribute NTP configuration updates.
Step 3	**show** *application_name* **status** **Example:** `switch(config)# show ntp status`	(Optional) For the specified application, displays the CFS distribution status.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable CFS to distribute Call Home configurations:

```
switch(config)# ntp distribute
switch(config)# show ntp status
Distribution : Enabled
switch(config)# copy running-config startup-config
[########################################] 100%
```

Specifying a CFS Distribution Mode

You can specify and enable a CFS distribution mode (Ethernet or IPv4).

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **cfs** [**eth** | **ipv4**] **distribute**
3. **show cfs status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	
	Example:
	`switch#` **`config t`**
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`cfs`** `[`**`eth`** `|` **`ipv4`**`]` **`distribute`**
	
	Example:
	`switch(config)# cfs ipv4 distribute`
	`switch(config)#`
		Globally enables CFS distribution over one of the following for all applications on the device.
		• Ethernet
		• IPv4
		In this example, CFS distribution is enabled over IPv4.

Step 3	**`show cfs status`**
	
	Example:
	`switch(config)#` **`show cfs status`**
	`Distribution : Enabled`
	`Distribution over IP : Enabled - mode IPv4`
	`IPv4 multicast address : 239.255.70.83`
	`switch(config)#`
		Shows the current state of CFS including distribution mode.
		In this example, CFS is shown as being distributed over IPv4.

Step 4	**`copy running-config startup-config`**
	
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring an IP Multicast Address for CFS Over IP

For CFS protocol-specific distributions, such as the keepalive mechanism for detecting network topology changes, use the IP multicast address to send and receive information.

You can configure the IP multicast address used to distribute CFS over IP for either of the following:

- IPv4—The default IPv4 multicast address is 239.255.70.83.

BEFORE YOU BEGIN

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

You must disable CFS IP distribution before changing the multicast address.

SUMMARY STEPS

1. **config t**
2. **no cfs** [**ipv4**] **distribute**
3. **cfs** [**ipv4**] **mcast-address** *ip_address*

4. **show cfs status**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	`Example``:`
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`no cfs`** `[`**`ipv4`**`]` **`distribute`**
	
	`Example``:`
	`switch(config)# no cfs ipv4 distribute`
	`This will prevent CFS from distributing`
	`over IPv4 network.`
	`Are you sure? (y/n)  [n] y`
	`switch(config)#`
		Globally disables CFS over IP distribution for all applications on the device.
		Note CFS over IP must be disabled before you can change the multicast address.

Step 3	**`cfs [ipv4] mcast-address`** *`ip_address`*
	
	`Example``:`
	`switch(config)# cfs ipv4 mcast-address`
	`239.255.1.1`
	`Distribution over this IP type will be`
	`affected`
	`Change multicast address for CFS-IP ?`
	`Are you sure? (y/n)  [n] y`
		Configures the multicast address for CFS distribution over IPv4. The ranges of valid IPv4 addresses are 239.255.0.0 through 239.255.255.255 and 239.192/16 through 239.251/16. The default IPv4 address is 239.255.70.83.

Step 4	**`show cfs status`**
	
	`Example``:`
	`switch(config)# show cfs status`
	`Distribution : Enabled`
	`Distribution over IP : Enabled - mode IPv4`
	`IPv4 multicast address : 239.255.1.1`
	`switch(config)#`
		Shows the current state of CFS including whether it is enabled, its IP mode, and its multicast addresses.
		In this example, CFS is shown as being distributed over IPv4 on 239.255.1.1.

Step 5	**`copy running-config startup-config`**
	
	`Example:`
	`switch(config)# copy running-config`
	`startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring CFS Regions

This section describes how to create and configure a CFS region and includes the following topics:

- Creating a CFS Region, page 2-13
- Moving an Application to a Different Region, page 2-14
- Removing an Application from a Region, page 2-15
- Deleting a CFS Region, page 2-16

Creating a CFS Region

You can create a CFS region and add an application, such as Call Home, to it.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **cfs region** *region_number*
3. *application_name*
4. **show cfs region brief**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**cfs region** *region_number*
	
	Example:
	`switch(config)# cfs region 4`
	`switch(config-cfs-region)#`
		Creates the region and places you into Configuration mode for the specified region.
		In this example, region 4 is created.

Step 3	*application_name*
	
	Example:
	`switch(config-cfs-region)# callhome`
	`switch(config-cfs-region)#`
		For the specified region, adds the named applications.

Step 4	**show cfs region brief**
	
	Example:
	`switch(config-cfs-region)# show cfs region brief`
	
	`---------------------------------------`
	`Region         Application   Enabled`
	`---------------------------------------`
	`4            callhome      yes`
	
	`switch(config-cfs-region)#`
		(Optional) Shows all configured regions and applications (does not show peers).
		In this example, the Call Home application is shown in region 4.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Moving an Application to a Different Region

You can move an application to a different region, for example, you can move NTP from region 1 to region 2.

Note When an application is moved, its scope is restricted to the new region; it ignores all other regions for distribution or merging purposes.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **cfs region** *region_number*
3. *application_name*
4. **show cfs region**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`cfs region`** *`region_number`*
	Example:
	`switch(config)# cfs region 2`
	`switch(config-cfs-region)#`
		Places you in configuration mode for the target/destination region.

Step 3	*`application_name`*
	Example:
	`switch(config-cfs-region)# callhome`
	`switch(config-cfs-region)# radius`
		Specifies applications to be moved.
		In this example, the Call Home application is moved to region 2.

Step 4	**`show cfs region name`** *`application_name`*
	Example:
	`switch(config-cfs-region)# show cfs region name callhome`
		Displays peers and region information for a given application.

Step 5	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to move the Call Home application to CFS region 2:

```
switch# config t
switch(config)# cfs region 2
switch(config-cfs-region)# callhome
switch(config-cfs-region)# show cfs region name callhome

Region-ID  : 2
Application: callhome
Scope      : Physical-fc-ip
-------------------------------------------------------------------------
 Switch WWN              IP Address
-------------------------------------------------------------------------
 20:00:00:22:55:79:a4:c1 172.28.230.85                           [Local]
                         switch

Total number of entries = 1

switch(config-cfs-region)#
```

Removing an Application from a Region

You can remove an application from a region. Removing an application from a region is the same as moving the application back to the default region, The default region is usually region 0. This action brings the entire fabric into the scope of distribution for the application.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **cfs region** *region_number*
3. **no** *application_name*
4. Repeat Step 3 for each application you want to remove from this region.
5. **show cfs region brief**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	`cfs region` *region_number*
	Example:
	`switch(config)# cfs region 2`
	`switch(config-cfs-region)#`
		Places you in Configuration mode for the specified region.

	Command	Purpose
Step 3	**no** *application_name* **Example:** `switch(config-cfs-region)# no ntp`	Removes the specified application from the region.
Step 4	(Optional) Repeat Step 3 for each application you want to remove from this region.	—

Step 5	**show cfs region brief**
	
	Example:
	`switch(config-cfs-region)# show cfs region brief`
	
	`---------------------------------------`
	` Region         Application   Enabled`
	`---------------------------------------`
	
	`   4            tacacs+       yes`
	`   6            radius        yes`
	
	`switch(config-cfs-region)#`
		Shows all configured regions and applications (does not show peers).

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Deleting a CFS Region

You can delete a region and move all included applications back to the default region.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **no cfs region** *region_number*
3. **show cfs region brief**
4. **show cfs application name** *application-name*
5. **copy running-config startup-config**

DETAILED STEPS

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td><pre>
config t

Example:
switch# config t
Enter configuration commands, one per
line. End with CNTL/Z.
switch(config)#
</pre></td><td>Places you in global configuration mode.</td></tr>
<tr><td>Step 2</td><td><pre>
no cfs region region_number

Example:
switch(config)# no cfs region 4
WARNING: All applications in the region
wiil be moved to default region.
Are you sure? (y/n) [n]
switch(config)#
</pre></td><td>Deletes the specified region after warning that this action causes all applications in the region to move to the default region.
After deleting the region, you are returned to global configuration mode.</td></tr>
<tr><td>Step 3</td><td><pre>
show cfs region brief

Example:
switch(config)# show cfs region brief

 Region   Application   Enabled
----
   6   radius   no

switch(config)#
</pre></td><td>Shows all configured regions and applications (does not show peers).
In this example, region 4 is absent.</td></tr>
<tr><td>Step 4</td><td><pre>
show cfs application name
application-name

Example:
switch# show cfs application name
callhome

 Enabled   : Yes
 Timeout   : 20s
 Merge Capable : Yes
 Scope   : Physical-fc-ip
 Region   : Default

switch#
</pre></td><td>Shows local application information by name.
In this case, the Call Home application is shown as now belonging to the default region.</td></tr>
<tr><td>Step 5</td><td><pre>
copy running-config startup-config

Example:
switch(config)# copy running-config
startup-config
</pre></td><td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td></tr>
</table>

Creating and Distributing a CFS Configuration

You can create a configuration change for an application and then distribute it to its application peers.

Caution If you do not commit the changes, they are not distributed and saved in the running configuration of application peer devices.

Caution If you do not save the changes to the startup configuration in every application peer device where distributed, then changes are retained only in their running configurations.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. *application_name*
3. *application_command*
4. Repeat Step 3 for each configuration command you want to make.
5. **show** *application_name* **status**
6. **commit**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	*`application_name`*
	Example:
	`switch(config)# callhome`
	`switch(config-callhome)#`
		Specifies that CFS starts a session for the specified application name and locks the fabric.

Step 3	*`application_command`*
	Example:
	`switch(config-callhome)# email-contact admin@Mycompany.com`
		Specifies that configuration changes are saved as a working copy and are not saved in the running configuration until you enter the **commit** command.

Step 4	(Optional) Repeat Step 3 for each configuration command you want to make.	—

	Command	Purpose

Step 5	**show** *application_name* **status**
	
	Example:
	`switch(config-callhome)# show callhome status`
	`Distribution : Enabled`
	`switch(config-callhome)#`
		(Optional) For the specified application, displays the CFS distribution status.
		In this example, the output shows that distribution is enabled for Call Home.

Step 6	**commit** **Example:** `switch(config-callhome)# commit`	CFS distributes the configuration changes to the running configuration of every application peer device. If one or more external devices report a successful status, the software overwrites the running configuration with the changes from the CFS working copy and releases the fabric lock. If none of the external devices report a successful status, no changes are made and the fabric lock remains in place.
Step 7	**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 in all devices in the fabric.

This example shows how to configure and distribute the contact information for Call Home:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server contact personname@companyname.com
switch(config)# callhome
switch(config-callhome)# email-contact admin@Mycompany.com
switch(config-callhome)# phone-contact +1-800-123-4567
switch(config-callhome)# street-address 123 Anystreet st. Anytown,AnyWhere
switch(config-callhome)# commit
switch(config-callhome)# copy running-config startup-config
[########################################] 100%
switch(config-callhome)#
```

Clearing a Locked Session

You can clear a lock held by an application from any device in the fabric.

You must have admin permissions to release a lock.

Caution When you clear a lock in the fabric, any pending configurations in any device in the fabric are discarded.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **show** *application_name* **status**
3. **clear** *application_name* **session**
4. **show** *application_name* **status**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Places you in global configuration mode.
Step 2	**show** *application_name* **status** switch(config)# show ntp status Distribution : Enabled Last operational state: Fabric Locked switch(config)#	Shows the current application state. In this example, NTP is shown as locked.
Step 3	**clear** *application_name* **session** **Example:** switch# clear ntp session switch#	Clears the application configuration session and releases the lock on the fabric. All pending changes are discarded.
Step 4	**show** *application_name* **status** **Example:** switch# show ntp status Distribution : Enabled Last operational state: No session switch#	Shows the current application state. This example shows that the lock is removed from the NTP application.

Discarding a Configuration

You can discard configuration changes and release the lock.

Caution If you discard configuration changes, the application flushes the pending database and releases locks in the fabric.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
1. *application_name* **abort**
2. **show** *application_name* **session status**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t switch(config)#	Places you in global configuration mode.
Step 2	switch(config)# *application_name* **abort** y **Example**: switch(config)# no cfs distribute This will prevent CFS from distributing the configuration to other switches. Are you sure? (y/n) [n] y	Aborts the application configuration after requesting confirmation. In this case, the NTP configuration is aborted, the changes to the configuration are discarded, the CFS session is closed, and the fabric lock is released. **Note** The **abort** command is supported only on the device where the fabric lock is acquired.
Step 3	**show** *application_name* **session status** **Example:** switch(config)# show ntp session status Last Action Time Stamp : Wed Nov 12 16:07:25 2008 Last Action : Abort Last Action Result : Success Last Action Failure Reason : none switch(config)#	(Optional) For the specified application, displays the CFS session status. In this example, the output shows that the CFS session was aborted.

Disabling CFS Distribution Globally

You can disable CFS distribution for a device, isolating the applications using CFS from fabric-wide distributions while maintaining physical connectivity.

When CFS is globally disabled on a device, CFS operations are restricted to the device and all CFS commands continue to function as if the device were physically isolated.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **no cfs distribute**
3. **show cfs status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	`Example:`
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	`switch(config)#` **`no cfs distribute`**
	
	`Example:`
	`switch(config)# no cfs distribute`
	`This will prevent CFS from distributing the`
	`configuration to other switches.`
	`Are you sure? (y/n)  [n] y`
	`switch(config)#`
		Globally disables CFS distribution for all applications on the device.
		Note If the virtual port channel (vPC) feature is enabled, then only IP distribution is disabled. You must first disable vPC before you can disable CFS distribution.

Step 3	**`show cfs status`**
	
	`Example:`
	`switch(config)# show cfs status`
	`Distribution : Enabled`
	`Distribution over IP : Disabled`
	`IPv4 multicast address : 239.255.70.83`
	`Distribution over Ethernet : Disabled`
	`switch(config)#`
		(Optional) Displays the global CFS distribution status (enabled/disabled) for the device.

Step 4	**`copy running-config startup-config`**
	
	`Example:`
	`switch(config)# copy running-config`
	`startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Verifying the CFS Configuration

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

Command	Purpose
show cfs application	Displays the applications that are currently CFS- enabled.
show cfs application name	Displays the details for a particular application, including enabled/disabled state, timeout as registered with CFS, merge capability if registered with CFS for merge support, distribution scope, and distribution region.
show *application_name* **session status**	Displays the configuration session status, including the last action, the result, and the reason if there was a failure.
show cfs internal	Displays information internal to CFS including memory statistics, event history, and so on.
show cfs lock	Displays all active locks.
show **cfs merge status** *name* [**detail**]	Displays the merge status for a given application.
show cfs peers	Displays all the peers in the physical fabric
show cfs regions	Displays all the applications with peers and region information.
show cfs static	Displays the status of all static peers.
show cfs status	Displays the status of CFS distribution on the device as well as IP distribution information.
show logging level cfs	Displays the CFS logging configuration.
show tech-support cfs	Displays information about the CFS configuration required by technical support when resolving a CFS issue.

Default Settings

Table 2-2 lists the default settings for CFS parameters.

Table 2-2 Default CFS Parameters

Parameters	Default
CFS distribution on the device	Enabled
CFS over IP	Disabled
IPv4 multicast address	239.255.70.83

Additional References

For additional information, see the following sections:

- Related Documents, page 2-24
- MIBs, page 2-24

Related Documents

Related Topic	Document Title
CFS CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
CFS configuration for Call Home	*Configuring Smart Call Home, page 6-1*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*
CFS configuration for TACACS+	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x*
CFS configuration for RADIUS	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x*
CFS configuration for roles	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x*

MIBs

MIBs	MIBs Link
• CISCO-CFS-MIB	*Cisco NX-OS MIB Support*

Feature History for CFS

This section provides the CFS release history.

Feature Name	Releases	Feature Information
CFS protocol	4.1(2)	This feature was introduced.

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:

Information About NTP

This section includes the following topics:

NTP Overview

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. With the User Datagram Protocol (UDP) as its transport protocol, NTP uses standard Universal Time Coordinated (UTC).

An NTP server usually receives its time from a 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 an atomic clock).
- A stratum 2 NTP server receives its time through NTP from a stratum 1 NTP 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 networking device to consider synchronizing with and to keep accurate time if a server failure occurs.

Distributing NTP Using CFS

Cisco Fabric Services (CFS) distributes the local NTP configuration to all Cisco devices in the network. After enabling CFS on your device, a network-wide lock is applied to NTP whenever an NTP configuration is started. After making the NTP configuration changes, you can discard or commit them. In either case, the CFS lock is then released from the NTP application.

For more information about CFS, see the "Configuring CFS" section on page 2-1.

High Availability

Stateless restarts are supported for NTP. After a reboot or a supervisor switchover, the running configuration is applied. For more information on high availability, see the *Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide*.

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

Virtualization Support

Up to one instance of NTP is supported on the entire platform. You must configure NTP in the default VDC. You are automatically placed in the default VDC unless you specify otherwise. For more information about VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

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 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.x* for more information about VRFs.

Licensing Requirements for NTP

Product	License Requirement
NX-OS	NTP requires no license and is bundled with the Cisco NX-OS system images at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *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.
- NTP must be configured in the default VDC. It cannot be configured in any other VDC except the default VDC.
- To configure VDCs, you must install the Advanced Services license. See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Guidelines and Limitations

NTP has the following configuration guidelines and limitations:

- NTP server functionality is not supported in Cisco NX-OS Release 4.2.
- You should have a peer association with another device only when you are sure that your clock is reliable (which means that you are a client of a reliable NTP server).
- A peer configured alone takes on the role of a server and should be used as a backup. If you have two servers, you can configure several devices to point to one server and the remaining devices to point to the other server. You can then configure a peer association between these two servers to create a more reliable NTP configuration.
- If you only have one server, you should configure all the devices as clients to that server.
- You can configure up to 64 NTP entities (servers and peers).
- If CFS is disabled for NTP, then NTP does not distribute any configuration and does not accept a distribution from other devices in the network.
- After CFS distribution is enabled for NTP, then the entry of an NTP configuration command locks the network for NTP configuration until a **commit** command is entered. During the lock, no changes can be made to the NTP configuration by any other device in the network except the device that initiated the lock.
- If you use CFS to distribute NTP, all devices in the network should have the same VRFs configured as you use for NTP.
- If you configure NTP in a VRF, ensure the NTP server and peers can reach each other through the configured VRFs.

Configuring NTP

This section includes the following topics:

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

Enabling or Disabling the NTP Protocol

You can enable or disable NTP. NTP is enabled by default. You can disable NTP and then reenable it.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **[no] ntp enable**
3. **show ntp status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**[no] ntp enable** **Example:** `switch(config)# ntp enable`	Enables or disables the NTP protocol on the entire device. NTP is enabled by default.

	Command	Purpose

Step 3	**show ntp status**
	Example:
	`switch(config)# show ntp status`
	`Distribution : Enabled`
	`Last operational state: Fabric Locked`
		(Optional) Displays the status of the NTP application.

Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to disable NTP:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# no ntp enable
```

Configuring an NTP Server and Peer

You can configure an NTP server and peer. You need to know the IP address or DNS names of your NTP server and its peers.

BEFORE YOU BEGIN

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

- If you plan to use CFS to distribute your NTP configuration to other devices, then you should have already completed the following:
  - Enable CFS distribution using the "Configuring CFS Distribution" section on page 2-5.
  - Enable CFS for NTP using the "Enabling CFS Distribution for NTP" section on page 3-7.

SUMMARY STEPS

1. **config t**
2. **ntp server** {*ip-address* | *ipv6-address* | *dns-name*} [**prefer**] [**use-vrf** *vrf-name*]
3. **ntp peer** {*ip-address* | *ipv6-address* | *dns-name*} [**prefer**] [**use-vrf** *vrf-name*]
4. **show ntp peers**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**ntp server** {*ip-address* | *ipv6-address* | *dns-name*} [**prefer**] [**use-vrf** *vrf-name*] **Example:** `switch(config)# ntp server 192.0.2.10`	Forms an association with a server. Optionally configures the NTP server to communicate over the specified VRF. The *vrf-name* can be any case-sensitive alphanumeric string up to 64 characters. Optionally use the **pefer** keyword to make this the preferred NTP server for the device.
Step 3	**ntp peer** {*ip-address* | *ipv6-address* | *dns-name*} [**prefer**] [**use-vrf** *vrf-name*] `switch(config)# ntp peer 2001:0db8::4101`	Forms an association with a peer. You can specify multiple peer associations. Optionally configures the NTP peer to communicate over the specified VRF. Optionally use the **pefer** keyword to make this the preferred NTP peer for the device. The *vrf-name* can be any case-sensitive alphanumeric string up to 64 characters.
Step 4	**show ntp peers** **Example:** `switch(config)# show ntp peers`	(Optional) Displays the configured server and peers. **Note** A domain name is resolved only when you have a DNS server configured.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure an NTP server and peer:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ntp server 192.0.2.105 use-vrf Red
switch(config)# ntp peer 2001:0db8::4101 use-vrf Red
switch(config)# show ntp peers
--------------------------------------------------
  Peer IP Address               Serv/Peer
--------------------------------------------------
  2001:db8::4101                Peer (configured)
  192.0.2.105                   Server (configured)
switch(config)# copy running-config startup-config
[########################################] 100%
switch(config)#
```

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 packet are sent. You can configure NTP to use a specific source IP address.

To configure the NTP source IP address, use the following command in global configuration mode:

Command	Purpose

`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.

Configuring the NTP Source Interface

You can configure NTP to use a specific interface.

To configure the NTP source interface, use the following command in global configuration mode:

Command	Purpose

`ntp source-interface` *`interf`*
`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.

Configuring NTP on a Secondary (Non-Default) VDC

You can configure a non-default VDC to get a timing update from the default VDC and its clients in order to synchronize with it.

BEFORE YOU BEGIN

Use the **switchto vdc** command to switch to the desired non-default VDC.

SUMMARY STEPS

1. **config t**
2. **feature ntp**
3. **ntp master**
4. (Optional) **ntp source-interface** *interface*
5. (Optional) **ntp source** *ip-address*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**feature ntp** **Example:** `switch(config)# feature ntp`	Enables NTP in the non-default VDC.
Step 3	**ntp master** **Example:** `switch(config)# ntp master`	Configures the device as an authoritative NTP server.
Step 4	**ntp source-interface** *interface* **Example:** `switch(config)# ntp source-interface ethernet 2/1`	(Optional) Configures the source interface for all NTP packets. Use the **?** keyword to display a list of supported interfaces.
Step 5	**ntp source** *ip-address* **Example:** `switch(config)# ntp source 192.0.2.1`	(Optional) Configures the source IP address for all NTP packets. The *ip-address* can be in IPv4 or IPv6 format.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Enabling CFS Distribution for NTP

You can enable CFS distribution for NTP in order to distribute the NTP configuration to other CFS-enabled devices.

BEFORE YOU BEGIN

- You have already enabled CFS distribution for the device using the "Configuring CFS Distribution" section on page 2-5.

SUMMARY STEPS

1. **configure terminal**
2. **ntp distribute**
3. **show ntp status**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Places you into CLI Global Configuration mode.
Step 2	switch(config)# **ntp distribute** **Example:** switch(config)# ntp distribute	Enables the device to receive NTP configuration updates that are distributed through CFS.
Step 3	**show ntp status** **Example:** switch(config)# show ntp status	(Optional) Displays the NTP CFS distribution status.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy run start [######################################] 100% switch(config)#	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Committing NTP Configuration Changes

When you commit the NTP configuration changes, the effective database is overwritten by the configuration changes in the pending database and all the devices in the network receive the same configuration.

To commit the NTP configuration changes, use the following command in global configuration mode:

Command	Purpose
ntp commit **Example**: switch(config)# ntp commit switch(config)#	Distributes the NTP configuration changes to all switches in the network and releases the CFS lock. Overwrites the effective database with the changes made to the pending database.

Discarding NTP Configuration Changes

After making the configuration changes, you can choose to discard the changes instead of committing them. If you discard the changes, Cisco NX-OS removes the pending database changes and releases the CFS lock.

To discard NTP configuration changes, use the following command in global configuration mode:

Command	Purpose
ntp abort **Example:** switch(config)# ntp abort	Discards the NTP configuration changes in the pending database and releases the CFS lock. Use this command on the device where you started the NTP configuration.

Releasing CFS Session Lock

If you have performed an NTP configuration and have forgotten to release the lock by either committing or discarding the changes, you or another administrator can release the lock from any device in the network. This will also discard pending database changes.

To release the session lock from any device and discard any pending database changes, use the following command in global configuration mode:

Command	Purpose
clear ntp session **Example:** switch(config)# clear ntp session	Discards the NTP configuration changes in the pending database and releases the CFS lock..

Verifying NTP Configuration

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

Command	Purpose
show ntp peer-status	Displays the status for all NTP servers and peers.
show ntp peers	Displays all the NTP peers.
show ntp pending peers	Displays the temporary CFS database for NTP.
show ntp pending-diff	Displays the difference between the pending CFS database and the current NTP configuration.
show ntp session status	Displays the NTP CFS distribution session information
show ntp statistics {**io** | **local** | **memory** | **peer** {**ipaddr** {*ipv4_addr* | *ipv6_addr*} | **name** peer_name}}	Displays the NTP statistics.
show ntp status	Displays the NTP CFS distribution status

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

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

NTP Example Configuration

This example shows how to configure an NTP server and peer and then save the configuration in startup so that it is saved across reboots and restarts:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ntp server 192.0.2.105
switch(config)# ntp peer 2001:0db8::4101
switch(config)# show ntp peers
--------------------------------------------------
  Peer IP Address               Serv/Peer
--------------------------------------------------
  2001:db8::4101                Peer (configured)
  192.0.2.105                   Server (configured)
switch(config)# copy running-config startup-config
[########################################] 100%
switch(config)#
```

Default Settings

Table 3-1 lists the default settings for NTP parameters.

Table 3-1 Default NTP Parameters

Parameters	Default
NTP	Enabled

Additional References

For additional information related to implementing NTP, see the following sections:

- Related Documents, page 3-11
- MIBs, page 3-12

Related Documents

Related Topic	Document Title
NTP CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

MIBs

MIBs	MIBs Link
- CISCO-NTP-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for NTP

Table 3-2 lists the release history for this feature.

Table 3-2 Feature History for NTP

Feature Name	Releases	Feature Information
CFS support	4.2(1)	Added ability to distribute NTP configuration using CFS. See the "Enabling CFS Distribution for NTP" section on page 3-8.
NTP source IP address or interface	4.1(3)	Added ability set the source IP address or source interface that NTP includes in all NTP packets sent to peers.
NTP protocol	4.0(3)	Added ability to disable the NTP protocol. See the "Enabling or Disabling the NTP Protocol" section on page 3-4.

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:

Information About CDP

This section includes the following topics:

CDP Overview

The Cisco Discovery Protocol (CDP) is a media- 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/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 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 4.x*.

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 details** command.

High Availability

Cisco NX-OS supports stateless restarts for CDP. After a reboot or a supervisor switchover, Cisco NX-OS applies the running configuration. For more information on high availability, see the *Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide*.

Virtualization Support

Product	License Requirement
NX-OS	Cisco NX-OS supports multiple instances of CDP, one instance per virtual device context (VDC). By default, Cisco NX-OS places you in the default VDC unless you specifically configure another VDC. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for CDP

CDP requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for CDP

CDP has the following prerequisites:

If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Guidelines and Limitations

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.

Configuring CDP

This section includes the following topics:

- Enabling or Disabling CDP Globally, page 4-4
- Enabling or Disabling CDP on an Interface, page 4-5
- Configuring Optional CDP Parameters, page 4-6

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

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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **cdp enable**
3. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	`Example:`
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`cdp enable`**
	
	`Example:`
	`switch(config)# cdp enable`
		Enables the CDP feature on the entire device. This is enabled by default.

Step 3	**`copy running-config startup-config`**
	
	`Example:`
	`switch(config)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Use the **no cdp enable** command to disable the CDP feature on the device.

Command	Purpose

`no cdp enable`
 	
`Example:`
`switch(config)# no cdp enable`
	Disables the CDP feature on the device.

This example shows how to enable the CDP feature:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# cdp enable
```

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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **interface** *interface-type slot/port*
3. **cdp enable**
4. **show cdp interface** *interface-type slot/port*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**interface** *interface-type slot/port* **Example:** switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	**cdp enable** **Example:** switch(config-if)# cdp enable	Enables CDP on this interface. This is enabled by default. **Note** Ensure that CDP is enabled on the device (see the "Enabling or Disabling CDP Globally" section on page 4-4).

	Command	Purpose
Step 4	**show cdp interface** *interface-type slot/port* **Example:** switch(config-if)# show cdp interface ethernet 1/2	(Optional) Displays CDP information for an interface.
Step 5	**copy running-config startup-config** **Example:** switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to disable CDP on Ethernet 1/2:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/2
switch(config-if)# no cdp enable
switch(config-if)# copy running-config startup-config
```

This example shows how to enable CDP on port channel 2:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface port-channel 2
switch(config-if)# cdp enable
switch(config-if)# copy running-config startup-config
```

Configuring Optional CDP Parameters

You can use the following optional commands in global configuration mode to modify CDP:

Command	Purpose
cdp advertise {**v1** | **v2**} **Example:** switch(config)# cdp advertise v1	Sets the CDP version supported by the device. The default is v2.
cdp format device-id {**mac-address** | **other** | **serial-number**} **Example:** switch(config)# cdp format device-id mac-address	Sets the CDP device ID. The options are as follows: • mac-address—MAC address of the chassis. • other—Chassis serial number • serial-number—Chassis serial number/Organizationally Unique Identifier (OUI) The default is other.

Command	Purpose

`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.

`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.

Verifying the CDP Configuration

Use the following commands to display the CDP configuration:

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-type slot/port*	Displays the CDP interface status.
show cdp neighbors {**device-id** | **interface** *interface-type slot/port*} [**detail**]	Displays the CDP neighbor status.
show cdp traffic interface *interface-type 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.

CDP Example Configuration

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

```
config t
cdp enable
cdp timer 50
cdp holdtime 100
```

Default Settings

Table 4-1 lists the CDP default settings.

Table 4-1 CDP Default Settings

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

Additional References

For additional information related to implementing CDP, see the following sections:

- Related Documents, page 4-8
- MIBs, page 4-8

Related Documents

Related Topic	Document Title
CDP CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

MIBs

MIBs	MIBs Link
• CISCO-CDP-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for CDP

Table 4-2 lists the release history for this feature.

Table 4-2 Feature History for Smart Call Home

Feature Name	Releases	Feature Information
CDP support for VTP domain name	4.2(1)	CDP advertises the VLAN Trunking Protocol (VTP) type-length-value field (TLV) in CDP version-2 packets. See VTP Feature Support, page 4-2.

CHAPTER 5

Configuring System Message Logging

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

This chapter includes the following sections:

Information 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.

System message logging is based on RFC 3164. 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. For information about configuring logging to terminal sessions, see the "Configuring System Message Logging to Terminal Sessions" section on page 5-3.

By default, the device logs system messages to a log file. For information about configuring logging to a file, see the "Logging System Messages to a File" section on page 5-5.

Table 5-1 describes the severity levels used in system messages. When you configure the severity level, the system outputs messages at that level and lower.

.

Table 5-1 System Message Severity Levels

Level	Description
0 – emergency	System unusable
1 – alert	Immediate action needed
2 – critical	Critical condition
3 – error	Error condition
4 – warning	Warning condition
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. For information about facilities, see the *Cisco Nexus 7000 Series NX-OS System Management Command Reference*. For information about configuring the severity level by module and facility, see the "Configuring Module and Facility Messages Logged" section on page 5-6.

This section includes the following topics:

- syslog Servers, page 5-2
- Virtualization Support, page 5-2

syslog Servers

The syslog servers run on remote systems that log system messages based on the syslog protocol. You can configure up to three IPv4 or IPv6 syslog servers. For information about configuring syslog servers, see the "Configuring syslog Servers" section on page 5-7.

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

Virtualization Support

A virtual device context (VDC) is a logical representation of a set of system resources. System message logging applies only to the VDC where commands are entered.

For information about configuring VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for System Message Logging

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

Guidelines and Limitations

System messages are logged to the console and the logfile by default.

Configuring System Message Logging

This section includes the following topics:

- Configuring System Message Logging to Terminal Sessions, page 5-3
- Logging System Messages to a File, page 5-5
- Configuring Module and Facility Messages Logged, page 5-6
- Configuring syslog Servers, page 5-7
- Displaying and Clearing Log Files, page 5-9

Note Be aware that the Cisco NX-OS commands for this feature may 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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **terminal monitor**
2. **config t**
3. **logging console** [*severity-level*]

   no logging console
4. **show logging console**

5. **logging monitor** [*severity-level*]

   no logging monitor
6. **show logging monitor**
7. **copy running-config startup-config**

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

Step 2	**config t**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 3	**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. Severity levels, which can range from 0 to 7, are listed in Table 5-1. If the severity level is not specified, the default of 2 is used.
	no logging console [*severity-level*] **Example:** `switch(config)# no logging console`	Disables the device's ability to log messages to the console.
Step 4	**show logging console** **Example:** `switch(config)# show logging console`	(Optional) Displays the console logging configuration.
Step 5	**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. The configuration applies to Telnet and SSH sessions. Severity levels, which can range from 0 to 7, are listed in Table 5-1. If the severity level is not specified, the default of 2 is used.
	no logging monitor [*severity-level*] **Example:** `switch(config)# no logging monitor`	Disables logging messages to Telnet and SSH sessions.
Step 6	**show logging monitor** **Example:** `switch(config)# show logging monitor`	(Optional) Displays the monitor logging configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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.

For information about displaying and clearing log files, see the "Displaying and Clearing Log Files" section on page 5-9.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **logging logfile** *logfile-name severity-level* [**size** *bytes*]

   no logging logfile [*logfile-name severity-level* [**size** *bytes*]]
3. **show logging info**
4. **copy running-config startup-config**

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**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. You can optionally specify a maximum file size. The default severity level is 5 and the file size is 10485760. Severity levels are listed in Table 5-1. The file size is from 4096 to 10485760 bytes.
	no logging logfile [*logfile-name severity-level* [**size** *bytes*]] **Example:** switch(config)# no logging logfile	Disables logging to the log file.
Step 3	**show logging info** **Example:** switch(config)# show logging info	(Optional) Displays the logging configuration.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring Module and Facility Messages Logged

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **logging module** [*severity-level*]
   no logging module
3. **show logging module**
4. **logging level** *facility severity-level*
   no logging level [*facility severity-level*]
5. **show logging level** [*facility*]
6. **logging timestamp** {**microseconds** | **milliseconds** | **seconds**}
   no logging timestamp {**microseconds** | **milliseconds** | **seconds**}
7. **show logging timestamp**
8. **copy running-config startup-config**

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**logging module** [*severity-level*] **Example:** switch(config)# logging module 3	Enables module log messages that have the specified severity level or higher. Severity levels, which range from 0 to 7, are listed in Table 5-1. If the severity level is not specified, the default of 5 is used.
	no logging module [*severity-level*] **Example:** switch(config)# no logging module	Disables module log messages.
Step 3	**show logging module** **Example:** switch(config)# show logging module	(Optional) Displays the module logging configuration.

	Command	Purpose
Step 4	**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. The facilities are listed in the *Cisco Nexus 7000 Series NX-OS System Management Command Reference*. Severity levels, which range from 0 to 7, are listed in Table 5-1. To apply the same severity level to all facilities, use the **all** facility. For defaults, see the **show logging level** command.
	no logging level [*facility severity-level*] **Example:** `switch(config)# no logging level aaa 3`	Resets the logging severity level for the specified facility to its default level. If you do not specify a facility and severity level, the device resets all facilities to their default levels.
Step 5	**show logging level** [*facility*] **Example:** `switch(config)# show logging level aaa`	(Optional) 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	**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.

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

Step 7	**show logging timestamp** **Example:** `switch(config)# show logging timestamp`	(Optional) Displays the logging time-stamp units configured.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring syslog Servers

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

Note We recommend that you configure the syslog server to use the management virtual routing and forwarding (VRF) instance. For more information on VRFs, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.x.*

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **logging server** *host* [*severity-level* [**use-vrf** *vrf-name*]]

   no logging server *host*
3. **show logging server**
4. **copy running-config startup-config**

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`logging server`** *`host`* `[`*`severity-level`*
	`[`**`use-vrf`** *`vrf-name`*`]]`
	
	Example 1:
	`switch(config)# logging server 192.0.2.253`
	
	Example 2:
	`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. In Cisco NX-OS Release 4.2 or higher, the default VRF is default. Severity levels, which range from 0 to 7, are listed in Table 5-1. The default outgoing facility is local7.
		
		Example 1 forwards all messages on facility local 7.
		
		Example 2 forwards messages with severity level 5 or lower for VRF red.

	`no logging server` *`host`*
	
	Example:
	`switch(config)# no logging server host`
		Removes the logging server for the specified host.

Step 3	**`show logging server`**
	
	Example:
	`switch(config)# show logging server`
		(Optional) Displays the syslog server configuration.

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

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
```

Table 5-2 describes the syslog fields that you can configure.

Table 5-2 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.
Action	Destination for messages, which can be a filename, a hostname preceded by the at sign (@), a comma-separated list of users, or an asterisk (*) for all logged-in users.

To configure a syslog server on a UNIX or Linux system, follow these 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 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.

BEFORE YOU BEGIN

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

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**

	Command	Purpose

Step 1	**`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 2007 nov 1 15:10:0`
		Displays the messages in the log file that have a time stamp 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 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 server	Displays the syslog server configuration.

show logging timestamp	Displays the logging time-stamp units configuration.
	Example:
	`switch(config)# show logging timestamp`
	`Logging timestamp:                Seconds`

For detailed information about the fields in the output from these commands, see the *Cisco Nexus 7000 Series NX-OS System Management Command Reference*.

System Message Logging Example Configuration

This example shows how to configure system message logging:

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

Default Settings

Table 5-3 lists the default settings for system message logging parameters.

Table 5-3 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; for severity levels, see the *Cisco Nexus 7000 Series NX-OS System Management Command Reference*.
Time-stamp units	Seconds
syslog server logging	Disabled

Additional References

For additional information related to implementing system message logging, see the following sections:

- Related Documents, page 5-12
- Standards, page 5-12

Related Documents

Related Topic	Document Title
System messages CLI commands	*Cisco NX-OS System Management Command Reference*
System messages	*Cisco NX-OS System Messages Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for System Message Logging

Table 5-4 lists the release history for this feature.

Table 5-4 Feature History for System Message Logging

Feature Name	Releases	Feature Information
IPv6 support	4.2(1)	Added support for IPv6 syslog hosts.
System Message Logging	4.0(1)	This feature was introduced.

CHAPTER 6

Configuring Smart Call Home

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

This chapter includes the following sections:

Information About Call Home

This section includes the following topics:

Call Home Overview

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

Call Home provides the following:

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

Destination Profiles

A destination profile includes the following information:

- One or more alert groups—The group of alerts that trigger a specific Call Home message if the alert occurs.
- One or more e-mail destinations—The list of recipients for the Call Home messages generated by alert groups assigned to this destination profile.
- Message format—The format for the Call Home message (short text, full text, or XML).
- Message severity level—The Call Home severity level that the alert must meet before Cisco NX-OS generates a Call Home message to all e-mail addresses in the destination profile. For more information about Call Home severity levels, see the “Call Home Message Urgency Levels” section on page 6-5. Cisco NX-OS does not generate an alert if the Call Home severity level of the alert is lower than the message severity level set for the destination profile.

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

Cisco NX-OS supports the following predefined destination profiles:

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

See the “Message Formats” section on page 6-25 for more information about the message formats.

Call Home Alert Groups

An alert group is a predefined subset of Call Home alerts that are supported in all Cisco NX-OS devices. Alert groups allow you to select the set of Call Home alerts that you want to send to a predefined or custom destination profile. Cisco NX-OS sends Call Home alerts to e-mail destinations in a destination profile only if that Call Home alert belongs to one of the alert groups associated with that destination profile and if the alert has a Call Home message severity at or above the message severity set in the destination profile (see the "Call Home Message Urgency Levels" section on page 6-5).

Table 6-1 lists supported alert groups and the default CLI command output included in Call Home messages generated for the alert group.

Table 6-1 Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Cisco-TAC	All critical alerts from the other alert groups destined for Smart Call Home.	Execute commands based on the alert group that originates the alert.
Configuration	Periodic events related to configuration.	**show module** **show running-configuration vdc-all all** **show startup-configuration vdc-all** **show vdc current** **show vdc membership** **show version**
Diagnostic	Events generated by diagnostics.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support gold** **show tech-support platform** **show tech-support sysmgr** **show vdc current** **show vdc membership** **show version**
EEM	Events generated by EEM.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show module** **show tech-support gold** **show tech-support platform** **show tech-support sysmgr** **show vdc current** **show vdc membership**
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	**show environment** **show logging last 200** **show module** **show vdc current** **show vdc membership** **show version**

Table 6-1 Alert Groups and Executed Commands (continued)

Alert Group	Description	Executed Commands
Inventory	Inventory status that is provided whenever a unit is cold booted, or when FRUs are inserted or removed. This alert is considered a noncritical event, and the information is used for status and entitlement.	**show inventory** **show license usage** **show module** **show system uptime** **show sprom all** **show vdc current** **show vdc membership** **show version**
License	Events related to licensing and license violations.	**show license usage vdc all** **show logging last 200** **show vdc current** **show vdc membership**
Linemodule hardware	Events related to standard or intelligent switching modules.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom alll** **show tech-support ethpm** **show tech-support gold** **show tech-support platform** **show tech-support sysmgr** **show vdc current** **show vdc membership** **show version**
Supervisor hardware	Events related to supervisor modules.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support platform** **show tech-support sysmgr** **show vdc current** **show vdc membership** **show version**
Syslog port group	Events generated by the syslog PORT facility.	**show license usage** **show logging last 200** **show vdc current** **show vdc membership**

Table 6-1 Alert Groups and Executed Commands (continued)

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

Call Home maps the syslog severity level to the corresponding Call Home severity level for syslog port group messages (see the "Call Home Message Urgency Levels" section on page 6-5).

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

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

Call Home Message Urgency Levels

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

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

Note Call Home does not change the syslog message level in the message text. The syslog messages in the Call Home log appear as they are described in the *Cisco NX-OS System Messages Reference.*

Table 6-2 lists each Call Home message level keyword and the corresponding syslog level for the syslog port alert group.

Table 6-2 Severity and syslog Level Mapping

Call Home Level	Keyword	syslog Level	Description
9	**Catastrophic**	N/A	Network-wide catastrophic failure.
8	**Disaster**	N/A	Significant network impact.
7	**Fatal**	Emergency (0)	System is unusable.
6	**Critical**	Alert (1)	Critical conditions that indicate that immediate attention is needed.

Table 6-2 Severity and syslog Level Mapping (continued)

Call Home Level	Keyword	syslog Level	Description
5	**Major**	Critical (2)	Major conditions.
4	**Minor**	Error (3)	Minor conditions.
3	**Warning**	Warning (4)	Warning conditions.
2	**Notification**	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	**Normal**	Information (6)	Normal event signifying return to normal state.
0	**Debugging**	Debug (7)	Debugging messages.

Obtaining Smart Call Home

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

Smart Call Home offers the following features:

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

You need the following information to register:

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

For more information about Smart Call Home, see the following Smart Call Home page:

http://www.cisco.com/go/smartcall/

Distributing Call Home Using CFS

You can use Cisco Fabric Services (CFS) to distribute a Call Home configuration to all CFS-enabled devices in the network. The entire Call Home configuration is distributed except the device priority and the sysContact names.

For more information about CFS, see the "Configuring CFS" section on page 2-1.

Database Merge Guidelines

When merging two Call Home databases, the following guidelines apply:

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

High Availability

Stateless restarts are supported for Call Home. After a reboot or supervisor switchover, the running configuration is applied.

Virtualization Support

One instance of Call Home is supported per virtual device context (VDC). Smart Call Home uses the contact information from the first registered VDC as the administrator contact for all VDCs on the physical device. For example, if you want the Smart Call Home to use the contact information from the default VDC, you should register using that VDC. You can update this information at the Smart Call Home web site at the following URL:

http://www.cisco.com/go/smartcall/

Smart Call Home registers the contacts for all other VDCs as users that can see all the Call Home data for the physical device but cannot act as administrators. All registered users and the registered administrator receive all Call Home notifications from all VDCs on the physical device.

By default, you are placed in the default VDC. In the default VDC, you can test Smart Call Home using the **callhome send** and **callhome test** commands. In a nondefault VDC, only the **callhome test** command is available. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

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

Licensing Requirements for Call Home

Product	License Requirement
NX-OS	Call Home requires no license, is bundled with the system images, and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for Call Home

Call Home has the following prerequisites:

- To send messages to an e-mail address, you must first configure an e-mail server. To send messages using HTTP, you must have access to an HTTPS server and have a valid certificate installed on the Nexus device.
- Your device must have IP connectivity to an e-mail server or HTTPS server.
- You must first configure the contact name (SNMP server contact), phone, and street address information. This step is required to determine the origin of messages received.
- If you use Smart Call Home, you need an active service contract for the device that you are configuring.
- If you configure VDCs, install the Advanced Services license (see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x)*. This license is required for VDCs only, not for Call Home.

Guidelines and Limitations

Call Home has the following configuration guidelines and limitations:

- If there is no IP connectivity or if the interface in the VRF to the profile destination is down, Call Home messages cannot be sent.
- Call Home operates with any SMTP server.
- Call Home messages sent through HTTP use the same VRF that you configure in the Call Home SMTP server.
- If you distribute the Call Home configuration using CFS, then the entire Call Home configuration is distributed except switch priority and the sysContact names.

Configuring Call Home

Note If you distribute the Call Home configuration using CFS, see the "Configuring Smart Call Home" section on page 6-1.

This section includes the following topics:

- Configuring Contact Information, page 6-9
- Creating a Destination Profile, page 6-11
- Modifying a Destination Profile, page 6-13
- Associating an Alert Group and a Destination Profile, page 6-15
- Adding show Commands to an Alert Group, page 6-17
- Configuring E-Mail, page 6-18
- Configuring Periodic Inventory Notifications, page 6-20
- Disabling Duplicate Message Throttle, page 6-21
- Enabling or Disabling Call Home, page 6-21
- Testing Call Home Communications, page 6-22

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

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

1. Configuring Contact Information, page 6-9
2. Creating a Destination Profile, page 6-11
3. Associating an Alert Group and a Destination Profile, page 6-15
4. (Optional) Adding show Commands to an Alert Group, page 6-17
5. (Optional) Creating and Distributing a CFS Configuration, page 2-19
6. Enabling or Disabling Call Home, page 6-21
7. (Optional) Testing Call Home Communications, page 6-22

Configuring Contact Information

You can configure the contact information for Call Home.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **snmp-server contact** *sys-contact*
3. **callhome**
4. **email-contact** *email-address*
5. **phone-contact** *international-phone-number*
6. **streetaddress** *address*
7. **contract-id** *contract-number*
8. **customer-id** *customer-number*

9. **site-id** *site-number*
10. **switch-priority** *number*
11. **commit**
12. **show callhome**
13. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

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

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

Step 5	**`phone-contact`** *`international-phone-number`*
	Example:
	`switch(config-callhome)# phone-contact +1-800-123-4567`
		Configures the phone number in international phone number format for the primary person responsible for the device. Up to 17 alphanumeric characters are accepted in international format.
		Note You cannot use spaces. Be sure to use the **+** prefix before the number.

Step 6	**`streetaddress`** *`address`*
	Example:
	`switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere`
		Configures the street address as an alphanumeric string with white spaces for the primary person responsible for the device. Up to 255 alphanumeric characters are accepted, including spaces.

Step 7	**`contract-id`** *`contract-number`*
	Example:
	`switch(config-callhome)# contract-id Contract5678`
		(Optional) Configures the contract number for this device from the service agreement. The contract number can be up to 255 alphanumeric characters in free format.

	Command	Purpose
Step 8	**customer-id** *customer-number* **Example:** `switch(config-callhome)# customer-id Customer123456`	(Optional) Configures the customer number for this device from the service agreement. The customer number can be up to 255 alphanumeric characters in free format.
Step 9	**site-id** *site-number* **Example:** `switch(config-callhome)# site-id Site1`	(Optional) Configures the site number for this device. The site number can be up to 255 alphanumeric characters in free format.
Step 10	**switch-priority** *number* **Example:** `switch(config-callhome)# switch-priority 3`	(Optional) Configures the switch priority for this device. The range is from 0 to 7, with 0 being the highest priority and 7 the lowest. The default is 7.
Step 11	**commit** **Example:** `switch(config-callhome)# commit`	Commits the callhome configuration commands.
Step 12	**show callhome** **Example:** `switch(config-callhome)# show callhome`	(Optional) Displays a summary of the Call Home configuration.
Step 13	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

This example shows how to configure the contact information for Call Home:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server contact personname@companyname.com
switch(config)# callhome
switch(config-callhome)# email-contact admin@Mycompany.com
switch(config-callhome)# phone-contact +1-800-123-4567
switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere
switch(config-callhome)# commit
```

Creating a Destination Profile

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

BEFORE YOU BEGIN

Make sure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **destination-profile** *name*

4. **destination-profile** *name* **format** {**XML** | **full-txt** | **short-txt**}
5. **commit**
6. **show callhome destination-profile** [**profile** *name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**callhome** **Example:** switch(config)# callhome switch(config-callhome)#	Enters callhome configuration mode.
Step 3	**destination-profile** *name* **Example:** switch(config-callhome)# destination-profile Noc101	Creates a new destination profile. The name can be any alphanumeric string up to 31 characters.
Step 4	**destination-profile** *name* **format** {**XML** | **full-txt** | **short-txt**} **Example:** switch(config-callhome)# destination-profile Noc101 format full-txt	Sets the message format for the profile. The name can be any alphanumeric string up to 31 characters.
Step 5	**commit** **Example:** switch(config-callhome)# commit	Commits the callhome configuration commands.
Step 6	**show callhome destination-profile** [**profile** *name*] **Example:** switch(config-callhome)# show callhome destination-profile profile Noc101	(Optional) Displays information about one or more destination profiles.
Step 7	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a destination profile for Call Home:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# callhome
switch(config-callhome)# destination-profile Noc101
switch(config-callhome)# destination-profile Noc101 format full-text
switch(config-callhome)# commit
```

Modifying a Destination Profile

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

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

See the "Associating an Alert Group and a Destination Profile" section on page 6-15 for information on configuring an alert group for a destination profile.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **destination profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **email-addr** *address*
4. **destination profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **http** *address*
5. **destination profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **transport-method** {**email** | **http**}
6. **destination profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-level** *number*
7. **destination profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-size** *number*
8. **commit**
9. **show call-home destination-profile** [**profile** *name*]
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

Step 3	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **email-addr** *address* **Example:** `switch(config-callhome)# destination-profile full-txt-destination email-addr person@place.com`	Configures an e-mail address for a user-defined or predefined destination profile. **Tip** You can configure up to 50 e-mail addresses in a destination profile.
Step 4	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **http** *address* **Example:** `switch(config-callhome)# destination-profile CiscoTAC-1 http http://site.com/service/callhome`	Configures an HTTP or HTTPS URL for a user-defined or predefined destination profile. The URL can be up to 255 characters. **Note** This command is not distributable with CFS. As a workaround, enter this command after the **commit** command.

Step 5	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **transport-method** {**email** | **http**}
	Example:
	`switch(config-callhome)# destination-profile CiscoTAC-1 http http://site.com/service/callhome`
		Configures an e-mail or HTTP transport method for a user-defined or predefined destination profile. The type of transport method that you choose determines the configured destination addresses of that type.
		Note This command is not distributable with CFS. As a workaround, enter this command after the **commit** command.

Step 6	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-level** *number* **Example:** `switch(config-callhome)# destination-profile full-txt-destination message-level 5`	Configures the Call Home message severity level for this destination profile. Cisco NX-OS sends only alerts that have a matching or higher Call Home severity level to destinations in this profile. The range is from 0 to 9, where 9 is the highest severity level.

	Command	Purpose
Step 7	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **message-size** *number* **Example:** switch(config-callhome)# destination-profile full-txt-destination message-size 100000	Configures the maximum message size for this destination profile The range is from 0 to 5000000. The default is 2500000.
Step 8	**commit** **Example:** switch(config-callhome)# commit	Commits the callhome configuration commands.
Step 9	**show callhome destination-profile** [**profile** *name*] **Example:** switch(config-callhome)# show callhome destination-profile profile full-text-destination	(Optional) Displays information about one or more destination profiles.
Step 10	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to modify a destination profile for Call Home:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# callhome
switch(config-callhome)# destination-profile full-text-destination email-addr
person@place.com
switch(config-callhome)# destination-profile full-text-destination message-level 5
switch(config-callhome)# destination-profile full-text-destination message-size 10000
switch(config-callhome)# commit
```

Associating an Alert Group and a Destination Profile

You can associate one or more alert groups with a destination profile.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **destination profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **alert-group** {**All** | **Cisco-TAC** | **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Linecard-Hardware** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test**}
4. **commit**

5. **show callhome destination-profile** [**profile** *name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**callhome** **Example:** switch(config)# callhome switch(config-callhome)#	Enters callhome configuration mode.
Step 3	**destination-profile** {*name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination**} **alert-group** {**All** | **Cisco-TAC** | **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Linecard-Hardware** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test**} **Example:** switch(config-callhome)# destination-profile Noc101 alert-group All	Associates an alert group with this destination profile. Use the **All** keyword to associate all alert groups with the destination profile.
Step 4	**commit** **Example:** switch(config-callhome)# commit	Commits the callhome configuration commands.
Step 5	**show callhome destination-profile** [**profile** *name*] **Example:** switch(config-callhome)# show callhome destination-profile profile Noc101	(Optional) Displays information about one or more destination profiles.
Step 6	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to associate all alert groups with the destination profile Noc101:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# callhome
switch(config-callhome)# destination-profile Noc101 alert-group All
switch(config-callhome)# commit
```

Adding show Commands to an Alert Group

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

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **alert-group** {**Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Linecard-Hardware** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test**} **user-def-cmd** *show-cmd*
4. **commit**
5. **show call-home user-def-cmds**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

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

	Command	Purpose
Step 5	**show callhome user-def-cmds** **Example:** `switch(config-callhome)# show callhome user-def-cmds`	(Optional) Displays information about all user-defined **show** commands added to alert groups.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

This example shows how to add the **show ip route** command to the Cisco-TAC alert group:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# callhome
switch(config-callhome)# alert-group Configuration user-def-cmd show ip route
switch(config-callhome)# commit
```

Configuring E-Mail

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **transport email smtp-server** *ip-address* [**port** *number*] [**use-vrf** *vrf-name*]
4. **transport email from** *email-address*
5. **transport email reply-to** *email-address*
6. **commit**
7. **show callhome transport-email**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**callhome** **Example:** switch(config)# callhome switch(config-callhome)#	Enters callhome configuration mode.
Step 3	**transport email smtp-server** *ip-address* [**port** *number*] [**use-vrf** *vrf-name*] **Example:** switch(config-callhome)# transport email smtp-server 192.0.2.1 use-vrf Red	Configures the SMTP server as either the domain name server (DNS) name, IPv4 address, or IPv6 address). Optionally configures the port number. The port ranges is from 1 to 65535. The default port number is 25. Also optionally configures the VRF to use when communicating with this SMTP server.
Step 4	**transport email from** *email-address* **Example:** switch(config-callhome)# transport email from person@company.com	(Optional) Configures the e-mail from field for Call Home messages.
Step 5	**transport email reply-to** *email-address* **Example:** switch(config-callhome)# transport email reply-to person@company.com	(Optional) Configures the e-mail reply-to field for Call Home messages.
Step 6	**commit** **Example:** switch(config-callhome)# commit	Commits the callhome configuration commands.
Step 7	**show callhome transport-email** **Example:** switch(config-callhome)# show callhome transport-email	(Optional) Displays information about the e-mail configuration for Call Home.
Step 8	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the e-mail options for Call Home messages:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# callhome
switch(config-callhome)# transport email smtp-server 192.0.2.10 use-vrf Red
switch(config-callhome)# transport email from person@company.com
switch(config-callhome)# transport email reply-to person@company.com
switch(config-callhome)# commit
```

Configuring Periodic Inventory Notifications

You can configure the switch to periodically send a message with an inventory of all software services currently enabled and running on the device along with hardware inventory information. Cisco NX-OS generates two Call Home notifications, periodic configuration messages and periodic inventory messages.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **callhome**
3. **periodic-inventory notification** [**interval** *days* | **timeofday** *time*]
4. **commit**
5. **show callhome**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

Step 3	**periodic-inventory notification** [**interval** *days*] [**timeofday** *time*] **Example:** `switch(config-callhome)# periodic-inventory notification interval 20`	Configures the periodic inventory messages. The interval range is from 1 to 30 days, and the default is 7. The *time* argument is in HH:MM format. It defines at what time of the day every *X* days an update is sent (where *X* is the update interval).
Step 4	**commit** **Example:** `switch(config-callhome)# commit`	Commits the callhome configuration commands.

	Command	Purpose
Step 5	**show callhome** **Example:** `switch(config-callhome)# show callhome`	(Optional) Displays information about Call Home.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

This example shows how to configure the periodic inventory messages to generate every 20 days:

```
switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# callhome
switch(config-callhome)# periodic-inventory notification interval 20
switch(config-callhome)# commit
```

Disabling Duplicate Message Throttle

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

Use the following commands in Call Home configuration mode to disable duplicate message throttling:

	Command	Purpose
Step 1	**no duplicate-message throttle** **Example:** `switch(config-callhome)# no duplicate-message throttle`	Disables duplicate message throttling for Smart Call Home. Enabled by default.
Step 2	**commit** **Example:** `switch(config-callhome)# commit`	Commits the callhome configuration commands.

Enabling or Disabling Call Home

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

Use the following commands in Call Home configuration mode to enable Call Home:

	Command	Purpose
Step 1	**enable** **Example:** `switch(config-callhome)# enable`	Enables Smart Call Home. Disabled by default. **Note** To disable Smart Call Home, use the **no enable** command in Smart Call Home configuration mode.
Step 2	**commit** **Example:** `switch(config-callhome)# commit`	Commits the callhome configuration commands.

Testing Call Home Communications

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

Use the following commands in any mode to generate a test Call Home message:

Command	Purpose
callhome send [**configuration** | **diagnostic**] **Example:** `switch(config-callhome)# callhome send diagnostic`	Sends the specified Call Home test message to all configured destinations. **Note** This command is available only in the default VDC.
callhome test **Example:** `switch(config-callhome)# callhome test`	Sends a test message to all configured destinations.

Verifying Call Home Configuration

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

Command	Purpose
show callhome	Displays the Call Home configuration.
show callhome destination-profile *name*	Displays one or more Call Home destination profiles.
show callhome merge	Displays the status of the last CFS merger for Call Home.
show callhome pending	Displays the Call Home configuration changes in the pending CFS database.
show callhome pending-diff	Displays the differences between the pending and running Call Home configuration.
show callhome session status	Displays the status of the last CFS commit or abort operation.

Command	Purpose
show callhome status	Displays the CFS distribution state (enabled or disabled) for Call Home.
show callhome transport-email	Displays the e-mail configuration for Call Home.
show callhome user-def-cmds	Displays CLI commands added to any alert groups.
show running-config callhome [**all**]	Displays the running configuration for Call Home.
show startup-config callhome	Displays the startup configuration for Call Home.
show tech-support callhome	Displays the technical support output for Call Home.

Call Home Example Configuration

This example shows how to create a destination profile called Noc101, associate the Cisco-TAC alert group to that profile, and configure contact and e-mail information:

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

Default Settings

Table 6-3 lists the default settings for Call Home parameters.

Table 6-3 Default Call Home Parameters

Parameters	Default
Destination message size for a message sent in full text format.	2,500,000
Destination message size for a message sent in XML format.	2,500,000
Destination message size for a message sent in short text format.	4000
SMTP server port number if no port is specified.	25

Table 6-3 Default Call Home Parameters (continued)

Parameters	Default
Alert group association with profile.	All for full-text-destination and short-text-destination profiles. The cisco-tac alert group for the CiscoTAC-1 destination profile.
Format type.	XML
Call Home message level.	0 (zero)

Additional References

For additional information related to implementing Call Home, see the following sections:

- Event Triggers, page 6-24
- Message Formats, page 6-25
- Sample syslog Alert Notification in Full-Text Format, page 6-29
- Sample syslog Alert Notification in XML Format, page 6-32
- Related Documents, page 6-35
- Standards, page 6-36
- MIBs, page 6-36

Event Triggers

Table 6-4 lists the event triggers and their Call Home message severity levels.

Table 6-4 Event Triggers

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

Table 6-4 Event Triggers (continued)

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

Message Formats

Call Home supports the following message formats:

- Short Text Message Format
- Common Fields for Full Text and XML Messages
- Inserted Fields for a Reactive and Proactive Event Message
- Inserted Fields for an Inventory Event Message
- Inserted Fields for a User-Generated Test Message

Table 6-5 describes the short text formatting option for all message types.

Table 6-5 Short Text Message Format

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event

Table 6-5 Short Text Message Format (continued)

Data Item	Description
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to system message

Table 6-6 describes the first set of common event message fields for full text or XML messages.

Table 6-6 Common Fields for Full Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Time stamp	Date and time stamp of event in ISO time notation: *YYYY-MM-DD HH:MM:SS GMT+HH:MM*.	/aml/header/time
Message name	Name of message. Specific event names are listed in Table 6-4.	/aml/header/name
Message type	Name of message type, such as reactive or proactive.	/aml/header/type
Message group	Name of alert group, such as syslog.	/aml/header/group
Severity level	Severity level of message (see the "Call Home Message Urgency Levels" section on page 6-5).	/aml/header/level
Source ID	Product type for routing, such as the Catalyst 6500 series switch.	/aml/header/source
Device ID	Unique device identifier (UDI) for the end device that generated the message. This field should be empty if the message is nonspecific to a device. The format is *type*@*Sid*@*serial*. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. An example is WS-C6509@C@12345678	/aml/ header/deviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header/customerID
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header /contractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	/aml/ header/siteId
Server ID	If the message is generated from the device, this is the unique device identifier (UDI) of the device. The format is *type*@*Sid*@*serial*. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is WS-C6509@C@12345678.	/aml/header/serverId

Table 6-6 Common Fields for Full Text and XML Messages (continued)

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

Table 6-7 describes the fields specific to alert group messages for full text and XML. These fields may be repeated if multiple CLI commands are executed for an alert group.

Table 6-7 Fields Specific to Alert Group Messages for Full Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Command output name	Exact name of the issued CLI command.	/aml/attachments/attachment/name
Attachment type	Specific command output.	/aml/attachments/attachment/type
MIME type	Either plain text or encoding type.	/aml/attachments/attachment/mime
Command output text	Output of command automatically executed (see the "Call Home Alert Groups" section on page 6-3).	/aml/attachments/attachment/atdata

Table 6-8 describes the reactive and proative event message format for full text or XML messages.

Table 6-8 Inserted Fields for a Reactive and Proactive Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
Affected FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name

Send document comments to nexus7k-docfeedback@cisco.com.

Table 6-8 Inserted Fields for a Reactive and Proactive Event Message (continued)

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

Table 6-9 describes the inventory event message format for full text or XML messages.

Table 6-9 Inserted Fields for an Inventory Event Message

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

Table 6-10 describes the user-generated test message format for full text or XML.

Table 6-10 Inserted Fields for a User-Generated Test Message

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

Sample syslog Alert Notification in Full-Text Format

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

```
Severity Level:5
Series:Nexus7000
Switch Priority:0
Device Id:N7K-C7010@C@TXX12345678
Server Id:N7K-C7010@C@TXX12345678
Time of Event:2008-01-17 16:31:33 GMT+0000 Message Name:
Message Type:syslog
System Name:dc3-test
Contact Name:Jay Tester
Contact Email:contact@example.com
Contact Phone:+91-80-1234-5678
Street Address:#1 Any Street
Event Description:SYSLOG_ALERT 2008 Jan 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR:
Error (0x20) while communicating with component MTS_SAP_ELTM
opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1)

syslog_facility:ETHPORT
start chassis information:
Affected Chassis:N7K-C7010
Affected Chassis Serial Number:TXX12345678 Affected Chassis Hardware Version:0.405
Affected Chassis Software Version:4.1(1) Affected Chassis Part No:73-10900-04 end chassis
information:
start attachment
    name:show logging logfile | tail -n 200
    type:text
    data:
    2008 Jan 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
cleared by user
    2008 Jan 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
    2008 Jan 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
    2008 Jan 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16:
Invalid argument: - sshd[14484]
    2008 Jan 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
    2008 Jan 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: "System Manager
(gsync controller)" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
    2008 Jan 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
    2008 Jan 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
    2008 Jan 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 3504)
hasn't caught signal 9 (no core).
    2008 Jan 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
    2008 Jan 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23210)
hasn't caught signal 9 (no core).
    2008 Jan 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
    2008 Jan 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23294)
hasn't caught signal 9 (no core).
    2008 Jan 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
    2008 Jan 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is
becoming active.
    2008 Jan 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed
- device_test
```

```
    2008 Jan 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR:  netstack [4336]
Unrecognized message from MRIB. Major type 1807
    2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 1
    2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 2
    2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 3
    2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 4
    2008 Jan 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2  -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10  -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2  -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0  -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0  -
ntpd[19045]
    2008 Jan 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET:  netstack [4336]  HA client
filter recovery failed (0)
    2008 Jan 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET:  netstack [4336]  HA client
filter recovery failed (0)
    2008 Jan 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
    2008 Jan 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
    2008 Jan 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
    2008 Jan 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
    2008 Jan 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
    2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 1
    2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 2
    2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 3
    2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 4
    2008 Jan 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
    2008 Jan 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
    2008 Jan 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present but
all AC inputs are not connected, ac-redundancy might be affected
    2008 Jan 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present but
all AC inputs are not connected, ac-redundancy might be affected
    2008 Jan 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
    2008 Jan 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
    2008 Jan 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 4820)
hasn't caught signal 9 (no core).
    2008 Jan 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
    2008 Jan 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24239)
hasn't caught signal 9 (no core).
```

```
    2008 Jan 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
    2008 Jan 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24401)
hasn't caught signal 9 (no core).
    2008 Jan 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
    2008 Jan 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
    2008 Jan 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24407)
hasn't caught signal 9 (no core).
    2008 Jan 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
    2008 Jan 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR:  netstack [4336]  (null)
    2008 Jan 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while
communicating with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP
(for:RID_PORT: Ethernet3/1) end attachment start attachment
    name:show vdc membership
    type:text
    data:

    vdc_id: 1 vdc_name: dc3-test interfaces:
        Ethernet3/1           Ethernet3/2           Ethernet3/3
        Ethernet3/4           Ethernet3/5           Ethernet3/6
        Ethernet3/7           Ethernet3/8           Ethernet3/9
        Ethernet3/10          Ethernet3/11          Ethernet3/12
        Ethernet3/13          Ethernet3/14          Ethernet3/15
        Ethernet3/16          Ethernet3/17          Ethernet3/18
        Ethernet3/19          Ethernet3/20          Ethernet3/21
        Ethernet3/22          Ethernet3/23          Ethernet3/24
        Ethernet3/25          Ethernet3/26          Ethernet3/27
        Ethernet3/28          Ethernet3/29          Ethernet3/30
        Ethernet3/31          Ethernet3/32          Ethernet3/33
        Ethernet3/34          Ethernet3/35          Ethernet3/36
        Ethernet3/37          Ethernet3/38          Ethernet3/39
        Ethernet3/40          Ethernet3/41          Ethernet3/42
        Ethernet3/43          Ethernet3/44          Ethernet3/45
        Ethernet3/46          Ethernet3/47          Ethernet3/48

    vdc_id: 2 vdc_name: dc3-aaa interfaces:

    vdc_id: 3 vdc_name: dc3-rbac interfaces:

    vdc_id: 4 vdc_name: dc3-call interfaces:



end attachment
start attachment
    name:show vdc current-vdc
    type:text
    data:
    Current vdc is 1 - dc3-test
end attachment
start attachment
    name:show license usage
    type:text
    data:
    Feature                      Ins  Lic   Status Expiry Date Comments
                                     Count
    --------------------------------------------------------------------------------
    LAN_ADVANCED_SERVICES_PKG     Yes   -   In use Never       -
    LAN_ENTERPRISE_SERVICES_PKG   Yes   -   Unused Never       -
    --------------------------------------------------------------------------------
end attachment
```

Sample syslog Alert Notification in XML Format

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

```
<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.cisco.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>1004:TXX12345678:478F82E6</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.cisco.com/2005/05/callhome/syslog</aml-block:Type>
<aml-block:CreationDate>2008-01-17 16:31:33 GMT+0000</aml-block:CreationDate>
<aml-block:Builder> <aml-block:Name>DC3</aml-block:Name>
<aml-block:Version>4.1</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>1005:TXX12345678:478F82E6</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>5</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.cisco.com/2005/05/callhome" version="1.0">
<ch:EventTime>2008-01-17 16:31:33 GMT+0000</ch:EventTime>
<ch:MessageDescription>SYSLOG_ALERT 2008 Jan 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR:
Error (0x20) while communicating with component MTS_SAP_ELTM
opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1) </ch:MessageDescription>
<ch:Event> <ch:Type>syslog</ch:Type> <ch:SubType></ch:SubType> <ch:Brand>Cisco</ch:Brand>
<ch:Series>Nexus7000</ch:Series> </ch:Event> <ch:CustomerData> <ch:UserData>
<ch:Email>contact@example.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:DeviceId>N7K-C7010@C@TXX12345678</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>dc3-test</ch:Name>
<ch:Contact>Jay Tester</ch:Contact> <ch:ContactEmail>contact@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber>+91-80-1234-5678</ch:ContactPhoneNumber>
<ch:StreetAddress>#1, Any Street</ch:StreetAddress> </ch:SystemInfo> </ch:CustomerData>
<ch:Device> <rme:Chassis xmlns:rme="http://www.cisco.com/rme/4.1">
<rme:Model>N7K-C7010</rme:Model>
<rme:HardwareVersion>0.405</rme:HardwareVersion>
<rme:SerialNumber>TXX12345678</rme:SerialNumber>
```

```
</rme:Chassis>
</ch:Device>
</ch:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging logfile | tail -n 200</aml-block:Name> <aml-block:Data
encoding="plain">
<![CDATA[2008 Jan 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
cleared by user
2008 Jan 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2008 Jan 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2008 Jan 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16: Invalid
argument: - sshd[14484]
2008 Jan 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2008 Jan 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: \"System Manager
(gsync controller)\" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
2008 Jan 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2008 Jan 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 3504)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23210)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23294)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
2008 Jan 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is becoming
active.
2008 Jan 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed -
device_test
2008 Jan 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR:  netstack [4336]  Unrecognized
message from MRIB. Major type 1807
2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 1
2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 2
2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 3
2008 Jan 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN in vdc 4
2008 Jan 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2  - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10  - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2  - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0  - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0  - ntpd[19045]
2008 Jan 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET:  netstack [4336]  HA client filter
recovery failed (0)
2008 Jan 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET:  netstack [4336]  HA client filter
recovery failed (0)
2008 Jan 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
2008 Jan 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
```

```
2008 Jan 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
2008 Jan 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
2008 Jan 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 1
2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 2
2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 3
2008 Jan 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP in vdc 4
2008 Jan 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
2008 Jan 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
2008 Jan 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present but all
AC inputs are not connected, ac-redundancy might be affected
2008 Jan 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present but all
AC inputs are not connected, ac-redundancy might be affected
2008 Jan 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
2008 Jan 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 4820)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24239)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24401)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
2008 Jan 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message Core
not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2008 Jan 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24407)
hasn&apos;t caught signal 9 (no core).
2008 Jan 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2008 Jan 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR:  netstack [4336]  (null)
2008 Jan 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT:
Ethernet3/1) ]]> </aml-block:Data> </aml-block:Attachment> <aml-block:Attachment
type="inline"> <aml-block:Name>show vdc membership</aml-block:Name> <aml-block:Data
encoding="plain"> <![CDATA[
vdc_id: 1 vdc_name: dc3-test interfaces:
    Ethernet3/1           Ethernet3/2           Ethernet3/3
    Ethernet3/4           Ethernet3/5           Ethernet3/6
    Ethernet3/7           Ethernet3/8           Ethernet3/9
    Ethernet3/10          Ethernet3/11          Ethernet3/12
    Ethernet3/13          Ethernet3/14          Ethernet3/15
    Ethernet3/16          Ethernet3/17          Ethernet3/18
    Ethernet3/19          Ethernet3/20          Ethernet3/21
    Ethernet3/22          Ethernet3/23          Ethernet3/24
    Ethernet3/25          Ethernet3/26          Ethernet3/27
    Ethernet3/28          Ethernet3/29          Ethernet3/30
    Ethernet3/31          Ethernet3/32          Ethernet3/33
    Ethernet3/34          Ethernet3/35          Ethernet3/36
    Ethernet3/37          Ethernet3/38          Ethernet3/39
    Ethernet3/40          Ethernet3/41          Ethernet3/42
    Ethernet3/43          Ethernet3/44          Ethernet3/45
    Ethernet3/46          Ethernet3/47          Ethernet3/48

vdc_id: 2 vdc_name: dc3-aaa interfaces:

vdc_id: 3 vdc_name: dc3-rbac interfaces:

vdc_id: 4 vdc_name: dc3-call interfaces:



]]>
</aml-block:Data>
</aml-block:Attachment>
<aml-block:Attachment type="inline">
<aml-block:Name>show vdc current-vdc</aml-block:Name> <aml-block:Data encoding="plain">
<![CDATA[Current vdc is 1 - dc3-test ]]> </aml-block:Data> </aml-block:Attachment>
<aml-block:Attachment type="inline"> <aml-block:Name>show license usage</aml-block:Name>
<aml-block:Data encoding="plain">
<![CDATA[Feature                      Ins  Lic   Status Expiry Date Comments
                                 Count
--------------------------------------------------------------------------------
LAN_ADVANCED_SERVICES_PKG     Yes   -   In use Never       -
LAN_ENTERPRISE_SERVICES_PKG   Yes   -   Unused Never       -
--------------------------------------------------------------------------------
]]>
</aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

Related Documents

Related Topic	Document Title
Call Home CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• CISCO-CALLHOME-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for Smart Call Home

Table 6-11 lists the release history for this feature.

Table 6-11 Feature History for Smart Call Home

Feature Name	Releases	Feature Information
Destination Profile Configuration	4.1(3)	The commands **destination profile http** and **destination profile transport-method** cannot be distributed. See the "Modifying a Destination Profile" section on page 6-13.

CHAPTER 7

Configuring Rollback

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

This chapter includes the following sections:

Information About Rollback

This section includes the following topics:

Rollback Overview

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

Cisco NX-OS automatically creates system checkpoints as described in the "Automatically Generated System Checkpoints" section on page 7-2. You can use either a user or system checkpoint to perform a rollback.

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

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

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

The default rollback type is atomic.

When you are ready to roll back to a checkpoint configuration, you can view the changes that will be applied to your current running configuration before committing to the rollback operation. If an error occurs during the rollback operation, you can choose to cancel the operation, or ignore the error and proceed with the rollback. If you cancel the operation, Cisco NX-OS provides a list of changes already applied before the error occurred. You need to clean up these changes manually.

Automatically Generated System Checkpoints

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

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

If one of these events causes system configuration changes, the feature software creates a system checkpoint that you can use to roll back to the previous system configuration.

The system generated checkpoint file names begin with “system-” and include the feature name. For example, the first time that you disable the EIGRP feature, the system creates the checkpoint named system-fm-__inst_1__eigrp.

High Availability

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

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

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

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

Virtualization Support

Cisco NX-OS creates a checkpoint of the running configuration in the virtual device context (VDC) that you are logged into. You can create different checkpoint copies in each VDC. You cannot apply the checkpoint of one VDC into another VDC. By default, Cisco NX-OS places you in the default VDC. See the *Cisco NX-OS Virtual Device Context Configuration Guide*.

VDC configuration does not support checkpoints for any operations, including (but not limited to) VDC creation, VDC deletion, VDC suspension, VDC reloading, VDC renaming, VDC interface allocation, shared interface allocation, FCoE VLAN allocation, resource allocation, and resource templates. You should create your checkpoint from within a specific VDC.

Licensing Requirements

Product	License Requirement
NX-OS	The rollback feature requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for Rollback

If you configure VDCs, install the Advanced Services license and go to the specific VDC. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

To configure the rollback feature, you must have network-admin or vdc-admin user privileges.

Guidelines and Limitations

Rollback has the following configuration guidelines and limitations:

- You can create up to ten checkpoint copies per VDC.
- You cannot apply the checkpoint file of one VDC into another VDC
- You cannot apply a checkpoint configuration in a nondefault VDC if there is a change in the global configuration portion of the running configuration compared to the checkpoint configuration.
- Your checkpoint file names must be 75 characters or less.
- You cannot start a checkpoint filename with the word *system*.
- Beginning in Cisco NX-OS Release 4.2(1), you can start a checkpoint filename with the word *auto*.
- Beginning in Cisco NS-OS Release 4.2(1), you can name a checkpoint file *summary* or any abbreviation of the word *summary*.
- Only one user can perform a checkpoint, rollback, or copy the running configuration to the startup configuration at the same time in a VDC.
- After the system executes the **write erase** or **reload** command, checkpoints are deleted. You can use the **clear checkpoint database** command to clear out all checkpoint files.

- A rollback fails for NetFlow if during a rollback, you try to modify a record that is programmed in the hardware.
- Although rollback is not supported for checkpoints across software versions, users can perform rollback at their own discretion and can use the best-effort mode to recover from errors.
- When checkpoints are created on bootflash, differences with the running-system configuration cannot be performed before performing the rollback, and the system reports “No Changes.”
- Checkpoints are local to a virtual device context (VDC).
- Checkpoints created using the **checkpoint** and **checkpoint** *checkpoint_name* commands are present upon a switchover for all VDCs.
- Checkpoints created in the default VDC are present upon reload unless a **write-erase** command is issued before a reload.
- Checkpoints created in nondefault VDCs are present upon reload only if a **copy run start** command is issued in the applicable VDC *and* the default VDC.
- Rollback to files on bootflash is supported only on files created using the **checkpoint** *checkpoint_name* command and not on any other type of ASCII file.
- Checkpoint names must be unique. You cannot overwrite previously saved checkpoints with the same name.

Configuring Rollback

This section includes the following topics:

- Creating a Checkpoint, page 7-4
- Implementing a Rollback, page 7-5

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

Creating a Checkpoint

You can create up to ten checkpoints of your configuration per VDC.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **checkpoint** {[*cp-name*] [**description** *descr*] | **file** *file-name*}

   no checkpoint *cp-name*
2. **show checkpoint** *cp-name* [**all**]

DETAILED STEPS

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

Implementing a Rollback

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

For information about automatically generated system checkpoints, see the "Automatically Generated System Checkpoints" section on page 7-2.

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

BEFORE YOU BEGIN

You are logged in to the device in EXEC mode for the correct VDC. To go to the correct VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. **show diff rollback-patch** {**checkpoint** *src-cp-name* | **running-config** | **startup-config** | **file** *source-file*} {**checkpoint** *dest-cp-name* | **running-config** | **startup-config** | **file** *dest-file*}
2. **rollback running-config** {**checkpoint** *cp-name* | **file** *cp-file*} [**atomic** | **best-effort** | **stop-at-first-failure**]

DETAILED STEPS

	Command	Purpose

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

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

Verifying the Rollback Configuration

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

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

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

Rollback Example Configuration

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

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

Default Settings

Table 7-1 lists the default settings for rollback parameters.

Table 7-1 Default Rollback Parameters

Parameters	Default
rollback type	atomic

Additional References

For additional information related to implementing a rollback, see the following sections:

- Related Documents, page 7-8
- Standards, page 7-8

Related Documents

Related Topic	Document Title
Rollback CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
configuration files	*Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.x*
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for Rollback

Table 7-2 lists the release history for this feature.

Table 7-2 Feature History for Rollback

Feature Name	Releases	Feature Information
High Availability	4.2(1)	Checkpoint and rollback operations support high availability. See the "High Availability" section on page 7-2.
Guidelines and Limitations	4.2(1)	Checkpoint file naming conventions changed. Rollback to files on bootflash is supported only on files created using the **checkpoint** *checkpoint_name* command. See the "Guidelines and Limitations" section on page 7-3.
Automatically generated system checkpoints	4.2(1)	The software automatically generates a system checkpoint when disabling a feature or license expiration could cause loss of configuration information. See the "Automatically Generated System Checkpoints" section on page 7-2.
Guidelines and Limitations	4.1(3)	A rollback fails for NetFlow if during rollback, you try to modify a record that is programmed in the hardware. A rollback is not supported for checkpoints across software versions. See the "Guidelines and Limitations" section on page 7-3.

CHAPTER 8

Configuring Session Manager

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

This chapter includes the following sections:

Information About Session Manager

This section includes the following topics:

Session Manager Overview

Session Manager allows you to implement configuration changes in batch mode, using the following phases:

- Configuration session—Creates a list of commands that you want to implement in Session Manager mode.
- Validation—Provides a basic semantic check on your configuration. Cisco NX-OS returns an error if the semantic check fails on any part of the configuration.
- Verification—Verifies the configuration as a whole, based on the existing hardware and software configuration and resources. Cisco NX-OS returns an error if the configuration does not pass this verification phase.

- Commit—Cisco NX-OS verifies the complete configuration and applies the changes to the device. If a failure occurs, Cisco NX-OS reverts to the original configuration.
- Abort—Discards the configuration changes before implementation.

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

High Availability

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

Virtualization Support

By default, Cisco NX-OS places you in the default VDC. See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for Session Manager

Product	License Requirement
NX-OS	Session Manager requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for Session Manager

If you configure VDCs, install the Advanced Services license and go to the specific VDC. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

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

Guidelines and Limitations

Session Manager has the following configuration guidelines and limitations:

- Session Manager supports only the ACL and QoS features.
- You can create up to 32 configuration sessions per VDC.
- You cannot issue an in-service software upgrade (ISSU) if an active session is in progress. You must commit the session, save it, or abort it before issuing an ISSU.
- You can configure a maximum of 20,000 commands across all sessions in a VDC.

- You cannnot simultaneously execute configuration commands in more then one configuration session or configuration terminal mode. Parallel configurations (for example, one configuration session and one configuration terminal) may cause validation or verification failures in the configuration session.
- If an interface reloads while you are configuring that interface in a configuration session, Sesson Manager may accept the commands even though the interface is not present in the device at that time.

Configuring Session Manager

This section includes the following topics:

- Creating a Session, page 8-3
- Configuring ACLs in a Session, page 8-4
- Verifying a Session, page 8-5
- Committing a Session, page 8-5
- Saving a Session, page 8-5
- Discarding a Session, page 8-6

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

Creating a Session

You can create up to 32 configuration sessions.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **configure session** *name*
2. **show configuration session** [*name*]
3. **save** *location*

DETAILED STEPS

	Command	Purpose

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

Step 2	**show configuration session** [*name*] **Example:** `switch(config-s)# show configuration session myACLs`	(Optional) Displays the contents of the session.
Step 3	**save** *location* **Example:** `switch(config-s)# save bootflash:sessions/myACLs`	(Optional) Saves the session to a file. The location can be in bootflash:, slot0:, or volatile:

Configuring ACLs in a Session

You can configure ACLs within a configuration session.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **configure session** *name*
2. **ip access-list** *name*
3. **permit** *protocol source destination*
4. **interface** *interface-type number*
5. **ip access-group** *name* {**in** | **out**}
6. **show configuration session** [*name*]

DETAILED STEPS

	Command	Purpose

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

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

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

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

Step 5	**ip access-group** *name* {**in** | **out**}
	Example:
	`switch(config-s-if)# ip access-group acl1 in`
		Specifies the direction of traffic the access group is applied to.

Step 6	**show configuration session** [*name*] **Example:** `switch(config-s)# show configuration session myacls`	(Optional) Displays the contents of the session.

Verifying a Session

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

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

Committing a Session

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

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

Saving a Session

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

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

Discarding a Session

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

Command	Purpose

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

Verifying the Session Manager Configuration

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

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

Session Manager Example Configuration

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

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

Additional References

For additional information related to implementing Session Manager, see the following sections:

- Related Documents, page 8-7
- Standards, page 8-7

Related Documents

Related Topic	Document Title
Session Manager CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
configuration files	*Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.x*
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

CHAPTER 9

Configuring the Scheduler

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

This chapter includes the following sections:

Information About the Scheduler

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

- Quality of Service policy changes
- Data backup
- Saving a configuration

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

This section includes the following topics:

Scheduler Overview

The scheduler defines a job and its timetable as follows:

- Job—A routine task or tasks defined as a command list and completed according to a specified schedule.
- Schedule—The timetable for completing a job. You can assign multiple jobs to a schedule. A schedule is defined as either periodic or one-time only:
  - Periodic mode—A recurring interval that continues until you delete the job. You can configure the following types of intervals:

    Daily—Job is completed once a day.

    Weekly—Job is completed once a week.

    Monthly—Job is completed once a month.

    Delta—Job begins at the specified start time and then at specified intervals (days:hours:minutes).
  - One-time mode—Job is completed only once at a specified time.

Remote User Authentication

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

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

Logs

The scheduler maintains a log file containing the job output. If the size of the job output is greater than the size of the log file, then the output is truncated. For more information, see the "Defining the Scheduler Log File Size" procedure on page 9-5.

High Availability

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

Virtualization Support

Jobs are created in the virtual device context (VDC) that you are logged into. By default, Cisco NX-OS places you in the default VDC. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for the Scheduler

Product	License Requirement
NX-OS	The scheduler requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for the Scheduler

The scheduler has the following prerequisites:

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

Guidelines and Limitations

The scheduler has the following configuration guidelines and limitations:

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

Configuring the Scheduler

This section includes the following topics:

Enabling the Scheduler

You can enable the scheduler feature so that you can configure and schedule jobs.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **feature scheduler**
3. **show scheduler config**
4. **copy running-config startup-config**

DETAILED STEPS

:

	Command or Action	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`feature scheduler`**
	
	Example:
	`switch(config)# feature scheduler`
		Enables the scheduler in the current VDC.

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

Step 4	**`copy running-config startup-config`**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Defining the Scheduler Log File Size

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **scheduler logfile size** *value*
3. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`config t`
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

Step 3	`copy running-config startup-config`
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Remote User Authentication

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

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

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **scheduler aaa-authentication password** [**0** | **7**] *password*
3. **scheduler aaa-authentication username** *name* **password** [**0** | **7**] *password*
4. **show running-config | include "scheduler aaa-authentication"**
5. **copy running-config startup-config**

DETAILED STEPS

:

	Command or Action	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

Step 3	**`scheduler aaa-authentication username`** *`name`* **`password`** `[`**`0`** `|` **`7`**`]` *`password`*
	
	Example:
	`switch(config)# scheduler`
	`aaa-authentication username newuser`
	`password Z98y76X54b`
		Configures a clear text password for a remote user.

Step 4	**`show running-config | include "scheduler aaa-authentication"`**
	
	Example:
	`switch(config)# show running-config |`
	`include "scheduler aaa-authentication"`
		(Optional) Displays the scheduler password information.

Step 5	**`copy running-config startup-config`**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Defining a Job

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

Caution Once a job is defined, you cannot modify or remove a command. To change the job, you must delete it and create a new one.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **scheduler job name** *string*
3. *command1* ;[*command2* ;*command3* ;...]
4. **show scheduler job** [*name*]
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**scheduler job name** *string* **Example:** switch(config)# scheduler job name backup-cfg switch(config-job)	Creates a job and enters job configuration mode. This example creates a scheduler job named backup-cfg.
Step 3	*command1* **;**[*command2* **;***command3* **;**...] **Example:** switch(config-job)# cli var name timestamp $(TIMESTAMP) ;copy running-config bootflash:/$(SWITCHNAME)-cfg.$(timestamp) ;copy bootflash:/$(SWITCHNAME)-cfg.$(timestamp) tftp://1.2.3.4/ vrf management switch(config-job)#	Defines the sequence of commands for the specified job. You must separate commands with a space and a semicolon (for example, " ;"). This example creates a scheduler job that saves the running configuration to a file in bootflash and then copies the file from bootflash to a TFTP server. The file name is created using the current time stamp and switch name.
Step 4	**show scheduler job** [*name*] **Example**: switch(config-job)# show scheduler job	(Optional) Displays the job information.
Step 5	**copy running-config startup-config** **Example:** switch(config-job)# copy running-config startup-config	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Deleting a Job

You can delete a job from the scheduler.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **no scheduler job name** *string*
3. **show scheduler job** [*name*]
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.

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

Step 3	**show scheduler job** [*name*] **Example**: switch(config-job)# show scheduler job name configsave	(Optional) Displays the job information.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Defining a Timetable

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

If you do not specify the time for the **time** commands, the scheduler assumes the current time. For example, if the current time is March 24, 2008, 22:00 hours, then jobs are started as follows:

- For the **time start 23:00 repeat 4:00:00** command, the scheduler assumes a start time of March 24, 2008, 23:00 hours.
- For the **time daily 55** command, the scheduler assumes a start time every day at 22:55 hours.
- For the **time weekly 23:00** command, the scheduler assumes a start time every Friday at 23:00 hours.

- For the **time monthly 23:00** command, the scheduler assumes a start time on the 24th of every month at 23:00 hours.

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **scheduler schedule name** *string*
3. **job name** *string*
4. **time daily** *time*

   time weekly [[**dow:**] *HH:*]*MM*

   time monthly [[**dm:**] *HH:*] *MM*

   time start {**now repeat** *repeat-interval* | *delta-time* [**repeat** *repeat-interval*]}
5. **show scheduler schedule** [*name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**scheduler schedule name** *string* **Example:** switch(config)# scheduler schedule name weekendbackupqos switch(config-schedule)#	Creates a new schedule and places you in schedule configuration mode for that schedule.
Step 3	**job name** *string* **Example:** switch(config-schedule)# job name offpeakZoning	Associates a job with this schedule. You can add multiple jobs to a schedule.

	Command or Action	Purpose
Step 4	**time daily** *time* **Example:** `switch(config-schedule)# time daily 23:00`	Indicates the job starts every day at a designated time specified as HH:MM.
	time weekly [[*dow:*]*HH:*]*MM* **Example:** `switch(config-schedule)# time weekly Sun:23:00`	Indicates that the job starts on a specified day of the week. • Day of the week (dow) specified as one of the following: – An integer such as 1 = Sunday, 2 = Monday, and so on. – An abbreviation such as Sun = Sunday. The maximum length for the entire argument is 10.
	time monthly [[*dm:*]*HH:*]*MM* **Example:** `switch(config-schedule)# time monthly 28:23:00`	Indicates the job starts on a specified day each month (dm). If you specify either 29, 30, or 31, the job is started on the last day of each month.

	time start {**now repeat** *repeat-interval* | *delta-time* [**repeat** *repeat-interval*]}
	
	Example:
	`switch(config-schedule)# time start now repeat 48:00`
		Indicates the job starts periodically.
		The start-time format is [[[[yyyy:]mmm:]dd:]HH]:MM.
		• *delta-time*
		Specifies the amount of time to wait after the schedule is configured before starting a job.
		• **now**
		Specifies that the job starts now.
		• **repeat** *repeat-interval*
		Specifies the frequency at which the job is repeated
		In this example, the job starts immediately and repeats every 48 hours.

Step 5	**show scheduler config** **Example**: `switch(config)# show scheduler config`	(Optional) Displays the scheduler configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Clearing the Scheduler Log File

You can clear the scheduler log file.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **clear scheduler logfile**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`clear scheduler logfile`**
	Example:
	`switch(config)# clear scheduler logfile`
		Clears the scheduler log file.

Disabling the Scheduler

You can disable the scheduler feature.

BEFORE YOU BEGIN

The scheduler feature must be enabled before you can configure and schedule jobs.

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

SUMMARY STEPS

1. **config t**
2. **no feature scheduler**
3. **show scheduler config**
4. **copy running-config startup-config**

DETAILED STEPS

:

	Command or Action	Purpose

Step 1	**`config t`**
	
	`Example:`
	`switch# config t`
	`Enter configuration commands, one per line.`
	`End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`no feature scheduler`**
	
	`Example:`
	`switch(config)# no feature scheduler`
		Disables the scheduler in the current VDC.

Step 3	**`show scheduler config`**
	
	`Example:`
	`switch(config)# show scheduler config`
	`                      ^`
	`% Invalid command at '^' marker.`
	`switch(config)#`
		(Optional) Displays the scheduler configuration. In this example, the scheduler feature is disabled so the command is not recognized.

Step 4	**`copy running-config startup-config`**
	
	`Example:`
	`switch(config)# copy running-config`
	`startup-config`
		(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Verifying the Scheduler Configuration

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

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

Scheduler Example Configurations

This section includes the following topics:

- Creating a Scheduler Job, page 9-13
- Scheduling a Scheduler Job, page 9-13
- Displaying the Job Schedule, page 9-13
- Displaying the Results of Running Scheduler Jobs, page 9-13

Creating a Scheduler Job

This example shows how to create a scheduler job that saves the running configuration to a file in bootflash and then copies the file from bootflash to a TFTP server (the filename is created using the current time stamp and switch name):

```
switch# config t
  switch(config)# scheduler job name backup-cfg
    switch(config-job)# cli var name timestamp $(TIMESTAMP) ;copy running-config
bootflash:/$(SWITCHNAME)-cfg.$(timestamp) ;copy bootflash:/$(SWITCHNAME)-cfg.$(timestamp)
tftp://1.2.3.4/ vrf management
    switch(config-job)# end
  switch(config)#
```

Scheduling a Scheduler Job

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

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

Displaying the Job Schedule

This example shows how to display the job schedule:

```
switch# show scheduler schedule
Schedule Name       : daily
---------------------------
User Name           : admin
Schedule Type       : Run every day at 1 Hrs 00 Mins
Last Execution Time : Fri Jan 2 1:00:00 2009
Last Completion Time: Fri Jan 2 1:00:01 2009
Execution count     : 2
-----------------------------------------------
     Job Name            Last Execution Status
-----------------------------------------------
back-cfg                          Success (0)
switch#
```

Displaying the Results of Running Scheduler Jobs

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

```
switch# show scheduler logfile
Job Name       : back-cfg                          Job Status: Failed (1)
Schedule Name  : daily                             User Name : admin
Completion time: Fri Jan 1  1:00:01 2009
--------------------------------- Job Output ---------------------------------
`cli var name timestamp 2009-01-01-01.00.00`
```

```
`copy running-config bootflash:/$(HOSTNAME)-cfg.$(timestamp)`
`copy bootflash:/switch-cfg.2009-01-01-01.00.00 tftp://1.2.3.4/ vrf management `
copy: cannot access file '/bootflash/switch-cfg.2009-01-01-01.00.00'
==============================================================================
Job Name       : back-cfg                         Job Status: Success (0)
Schedule Name  : daily                            User Name : admin
Completion time: Fri Jan 2  1:00:01 2009
--------------------------------- Job Output ---------------------------------
`cli var name timestamp 2009-01-02-01.00.00`
`copy running-config bootflash:/switch-cfg.2009-01-02-01.00.00`
`copy bootflash:/switch-cfg.2009--01-02-01.00.00 tftp://1.2.3.4/ vrf management `
Connection to Server Established.
[                         ]         0.50KBTrying to connect to tftp server......
[######                   ]        24.50KB

TFTP put operation was successful
==============================================================================
switch#
```

Default Settings

Table 9-1 lists the scheduler default settings.

Table 9-1 Default Command Scheduler Parameters

Parameters	Default
Scheduler state	Disabled.
Log file size	16 KB.

Additional References

For additional information related to the scheduler, see the following sections:

- Related Documents, page 9-14
- Standards, page 9-15

Related Documents

Related Topic	Document Title
Scheduler CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x*
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

CHAPTER 10

Configuring SNMP

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

This chapter includes the following sections:

Information 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.

This section includes the following topics:

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. Cisco NX-OS 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.

Cisco NX-OS 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 a 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 (see the "Configuring SNMP Notification Receivers with VRFs" section on page 10-14). 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. Cisco NX-OS 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 Cisco NX-OS never receives a response, it can send the inform request again.

You can configure Cisco NX-OS to send notifications to multiple host receivers. See the "Configuring SNMP Notification Receivers" section on page 10-11 for more information about host receivers.

Table 10-1 lists the SNMP traps that are enabled by default.

Table 10-1 SNMP Traps Enabled By Default

Trap Type	Description
generic	: coldStart
generic	: warmStart
entity	: entity_mib_change
entity	: entity_module_status_change
entity	: entity_power_status_change
entity	: entity_module_inserted
entity	: entity_module_removed
entity	: entity_unrecognised_module

Table 10-1 SNMP Traps Enabled By Default (continued)

Trap Type	Description
entity	: entity_fan_status_change
entity	: entity_power_out_change
link	: linkDown
link	: linkUp
link	: extended-linkDown
link	: extended-linkUp
link	: cieLinkDown
link	: cieLinkUp
link	: delayed-link-state-change
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
rmon	: risingAlarm
rmon	: fallingAlarm
rmon	: hcRisingAlarm
rmon	: hcFallingAlarm
entity	: entity_sensor

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 as follows:

- Message integrity—Ensures that a packet has not been tampered with while it was in-transit.
- Authentication—Determines that 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.

This section includes the following topics:

- Security Models and Levels for SNMPv1, v2, v3, page 10-4
- User-Based Security Model, page 10-4
- CLI and SNMP User Synchronization, page 10-5

- Group-Based SNMP Access, page 10-6

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.
- 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.

Table 10-2 identifies what the combinations of security models and levels mean.

Table 10-2 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	noAuthNoPriv	Username	No	Uses a username 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 that uses 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 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 (password, roles, and so on).

Cisco NX-OS holds the synchronized user configuration for 60 minutes by default. See the "Modifying the AAA Synchronization Time" section on page 10-23 for information on how to modify this default value.

Group-Based SNMP Access

Note Because *group* is a standard SNMP term used industry-wide, we refer to role(s) as group(s) 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.

See Chapter 13, "Configuring the Embedded Event Manager" for more information about EEM.

Multiple Instance Support

A device can support multiple instances of a logical network entity, such as protocol instances or VRFs. 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 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.

To map an SNMP context to a logical network entity, follow these steps:

Step 1 Create the SNMPv3 context.

Step 2 Determine the logical network entity instance.

Step 3 Map the SNMPv3 context to a logical network entity.

Step 4 Optionally, map the SNMPv3 context to an SNMPv2c community.

For more information, see the "Configuring the Context to Network Entity Mapping" section on page 10-21.

High Availability

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

Virtualization Support

Cisco NX-OS supports one instance of the SNMP per virtual device context (VDC). By default, Cisco NX-OS places you in the default VDC. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

SNMP supports multiple MIB module instances and maps them to logical network entities. For more information, see the "Multiple Instance Support" section on page 10-6.

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. For more information, see the "Configuring SNMP Notification Receivers with VRFs" section on page 10-14).

Licensing Requirements for SNMP

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

Prerequisites for SNMP

SNMP has the following prerequisites:

- If you configure VDCs, install the Advanced Services license and enter the desired VDC. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x.*

Guidelines and Limitations

SNMP has the following configuration guidelines and limitations:

- Cisco NX-OS supports read-only access to some SNMP MIBs. See the Cisco NX-OS MIB support list at the following URL for more information:

  http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Configuring SNMP

This section includes the following topics:

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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in 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	**show snmp user** **Example:** switch(config-callhome)# show snmp user	(Optional) Displays information about one or more SNMP users.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the SNMP contact and location information:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp-server user Admin auth sha abcd1234 priv abcdefgh
```

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 authorizationError for any SNMPv3 PDU request using securityLevel parameter of either noAuthNoPriv or authNoPriv.

Use the following command in global configuration mode to enforce SNMP message encryption for a user:

Command	Purpose
snmp-server user *name* **enforcePriv** **Example:** `switch(config)# snmp-server user Admin enforcePriv`	Enforces SNMP message encryption for this user.

Use the following command in global configuration mode to enforce SNMP message encryption for all users:

Command	Purpose
snmp-server globalEnforcePriv **Example:** `switch(config)# snmp-server globalEnforcePriv`	Enforces SNMP message encryption for all users.

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.

Use the following command in global configuration mode to assign a role to an SNMP user:

Command	Purpose
snmp-server user *name group* **Example:** `switch(config)# snmp-server user Admin superuser`	Associates this SNMP user with the configured user role.

Creating SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

Use the following command in global configuration mode to create an SNMP community string:

Command	Purpose

snmp-server community *name group* {**ro** | **rw**}
Example:
`switch(config)# snmp-server community public ro`
	Creates an SNMP community string.

Filtering SNMP Requests

You can assign an access list (ACL) to a community to filter incoming 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)

See the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x* for more information on creating ACLs. The ACL applies to both IPv4 and IPv6 over UDP and TCP.

Use the following command in global configuration mode to assign an ACL to a community to filter SNMP requests:

Command	Purpose
snmp-server community *community-name* **use-acl** *acl-name* **Example:** switch(config)# snmp-server community public use-acl my_acl_for_public	Assigns an ACL to an SNMP community to filter SNMP requests.

Configuring SNMP Notification Receivers

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

Use the following command in global configuration mode to configure a host receiver for SNMPv1 traps:

Command	Purpose
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.

Use the following command in global configuration mode to configure a host receiver for SNMPv2c traps or informs:

Command	Purpose
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.

Use the following command in global configuration mode to configure a host receiver for SNMPv3 traps or informs:

Command	Purpose
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.

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 this 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.

Use the following command in global configuration mode to configure a host receiver on a source interface:

Command	Purpose
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.

Use the following command in global configuration mode to configure a source interface for sending out all SNMP notifications:

Command	Purpose
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.

Use the **show snmp source-interface** command to display 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.

Use the following command in global configuration mode to configure the notification target user:

Command	Purpose
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 engineID format is a 12-digit colon-separated decimal number.

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.

Use the following command in global configuration mode to configure a VRF to use for sending notifications to the host receiver:

Command	Purpose
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 thc ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.
no snmp-server host *ip-address* **use-vrf** *vrf_name* [**udp_port** *number*] **Example:** `switch(config)# no snmp-server host 192.0.2.1 use-vrf Blue`	Removes the VRF reachability information for the configured host, and removes the entry from thc ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. The *ip-address* can be an IPv4 or IPv6 address. Does not remove the host configuration.

You can configure Cisco NX-OS filter notifications based on the VRF in which the notification occurred.

Use the following command in global configuration mode to filter notifications based on a configured VRF:

Command	Purpose
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 thc ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

Command	Purpose

`no snmp-server host` *`ip-address`* **`filter-vrf`**
`vrf_name`
`Example:`
`switch(config)# no snmp-server host 192.0.2.1 filter-vrf Red`
	Removes the VRF filter information for the configured host, and removes the entry from thc ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.
	The *ip-address* can be an IPv4 or IPv6 address. This command does not remove the host 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. **config t**
2. **snmp-server source-interface traps** *if-type if-number*
3. **show snmp source-interface**
4. **snmp-server host** *ip-address* **use-vrf** *vrf_name* [**udp_port** *number*]
5. **show snmp host**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	`Example:`
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in 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 this command: **snmp-server host** *ip-address* **source-interface** *if-type if-number.*

	Command	Purpose
Step 3	**show snmp source-interface** **Example:** switch(config)# show snmp source-interface	(Optional) Displays information about configured source interfaces.
Step 4	**snmp-server host** *ip-address* **use-vrf** *vrf_name* [**udp_port** *number*] **Example:** switch(config)# snmp-server host 171.71.48.164 use_vrf default	Configures SNMP to use the selected VRF to communicate with the host receiver. The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB. **Note** By default, SNMP sends the traps using the management VRF. If you do not want to use the management VRF, you must use this command to specify the desired VRF.
Step 5	**show snmp host** **Example:** switch(config)# show snmp host	(Optional) Displays information about configured SNMP hosts.
Step 6	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

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

```
switch# config t
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
-------------------------------------------------------------------
```

Enabling SNMP Notifications

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

Table 10-3 lists the commands that enable the notifications for Cisco NX-OS MIBs.

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

Table 10-3 Enabling SNMP Notifications

MIB	Related Commands
All notifications	**snmp-server enable traps**
CISCO-AAA-SERVER-MIB	**snmp-server enable traps aaa**
CISCO-STP-BRIDGE-MIB	**snmp-server enable traps bridge**
CISCO-CALLHOME-MIB	**snmp-server enable traps callhome** **snmp-server enable traps callhome event-notify** **snmp-server enable traps callhome smtp-send-fail**
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp**
ENTITY-MIB, CISCO-ENTITY-FRU-CONTROL-MIB, CISCO-ENTITY-SENSOR-MIB	**snmp-server enable traps entity** **snmp-server enable traps entity fru**
CISCO-HSRP-MIB	**snmp-server enable traps hsrp** **snmp-server enable traps hsrp state-change**
CISCO-LICENSE-MGR-MIB	**snmp-server enable traps license**
IF-MIB	**snmp-server enable traps link**
CISCO-RF-MIB	**snmp-server enable traps rf**
SNMPv2-MIB	**snmp-server enable traps snmp** **snmp-server enable traps snmp authentication**
CISCO-STPX-MIB	**snmp-server enable traps stpx**

Use the following commands in global configuration mode 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.

Command	Purpose
snmp-server enable traps bridge [**newroot** | **topologychange**] **Example:** `switch(config)# snmp-server enable traps bridge newroot`	Enables the STP bridge SNMP notifications.
snmp-server enable traps callhome [**event-notify**] [**smtp-send-fail**] **Example:** `switch(config)# snmp-server enable traps callhome`	Enables the CISCO-CALLHOME-MIB SNMP notifications. Optionally, enables the following specific notifications: • **event-notify**—Enables Call Home external event notifications. • **smtp-send-fail**—Enables SMTP message send fail notifications.
snmp-server enable traps eigrp **Example:** `switch(config)# snmp-server enable traps eigrp`	Enables the CISCO-EIGRP-MIB SNMP notifications.
snmp-server enable traps entity [**fru**] **Example:** `switch(config)# snmp-server enable traps entity`	Enables the ENTITY-MIB SNMP notifications.
snmp-server enable traps hsrp [**state-change**] **Example:** `switch(config)# snmp-server enable traps hsrp`	Enables the HSRP-MIB SNMP notifications.
snmp-server enable traps license **Example:** `switch(config)# snmp-server enable traps license`	Enables the license SNMP notification.
snmp-server enable traps link **Example:** `switch(config)# snmp-server enable traps link`	Enables the link SNMP notifications.
snmp-server enable traps rf **Example:** `switch(config)# snmp-server enable traps rf`	Enables the redundancy framework (RF) SNMP notifications.
snmp-server enable traps [*trap-arg* [*trap_sub_category*] | **snmp** [**authentication**]] **Example:** `switch(config)# snmp-server enable traps snmp`	Enables the SNMP agent notifications.

Command	Purpose
snmp-server enable traps *trap-arg-global-scope* [*trap_sub_category*] **Example:** switch(config)# snmp-server enable traps entity	Enables the SNMP agent notifications at global scope.
snmp-server enable traps stpx [**inconsistency** | **loop-inconsistency** | **root-inconsistency**] **Example:** switch(config)# snmp-server enable traps stpx root-inconsistency	Enables the STPX SNMP notifications.

Disabling LinkUp/LinkDown Notifications on an Interface

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

Use the following command in interface configuration mode to disable linkUp/linkDown notifications for the interface:

Command	Purpose
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.

Displaying SNMP ifIndex for an Interface

The SNMP ifIndex is used across multiple SNMP MIBs to link related interface information. The ifIndex is also used by NetFlow to collect information on an interface.

Use the following command in any mode to display the SNMP ifIndex values for interfaces:

Command	Purpose
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 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.

Use the following command in global configuration mode to enable a one-time authentication for SNMP over TCP:

Command	Purpose
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.

Assigning the 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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **snmp-server contact** *name*
3. **snmp-server location** *name*
4. **show snmp**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in 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.

	Command	Purpose
Step 4	**show snmp** **Example:** switch(config)# show snmp	(Optional) Displays information about one or more destination profiles.
Step 5	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the SNMP contact and location information:

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# snmp contact Admin
switch(config)# snmp location Lab-7
```

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

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

Determine the logical network entity instance. For more information on VRFs and protocol instances, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.x*, or the *Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide, Release 4.x*.

SUMMARY STEPS

1. **config t**
2. **snmp-server context** *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*]
3. **snmp-server mib community-map** *community-name* **context** *context-name*
4. **show snmp context**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**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.
Step 3	**snmp-server mib community-map** *community-name* **context** *context-name* **Example:** `switch(config)# snmp-server mib community-map public context public1`	(Optional) Maps an SNMPv2c community to an SNMP context. The names can be any alphanumeric string up to 32 characters.

Step 4	**`show snmp context`**
	Example:
	`switch(config)# show snmp context`
		(Optional) Displays information about one or more SNMP contexts.

Step 5	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves this configuration change.

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

```
switch# config t
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# config t
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
```

Use the following command in global configuration mode to delete the mapping between an SNMP context and a logical network entity:

Command	Purpose
no snmp-server context *context-name* [**instance** *instance-name*] [**vrf** *vrf-name*] [**topology** *topology-name*] **Example:** `switch(config)# no snmp-server context public1`	Deletes the mapping between an SNMP context and a protocol instance, VRF, or topology. The names can be any alphanumeric string up to 32 characters. **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.

Disabling SNMP

You can disable SNMP on a device.

Use the following command in global configuration mode to disable SNMP:

Command	Purpose
no snmp-server protocol enable **Example:** `switch(config)# no snmp-server protocol enable`	Disables SNMP. This command is enabled by default.

Modifying the AAA Synchronization Time

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

Use the following command in global configuration mode to modify the AAA synchronization time:

Command	Purpose
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.

Verifying SNMP Configuration

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

Command	Purpose
show interface snmp-ifidex	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.
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.

SNMP Example Configurations

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:

```
config t
 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# config t
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
-------------------------------------------------------------------
```

Default Settings

Table 10-4 lists the default settings for SNMP parameters.

Table 10-4 Default SNMP Parameters

Parameters	Default
license notifications	Enabled.

Additional References

For additional information related to implementing SNMP, see the following sections:

- Related Documents, page 10-25
- Standards, page 10-26
- MIBs, page 10-26

Related Documents

Related Topic	Document Title
SNMP CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*
MIBs	http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• SNMP-COMMUNITY-MIB • SNMP-FRAMEWORK-MIB • SNMP-NOTIFICATION-MIB • SNMP-TARGET-MIB • SNMPv2-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for SNMP

Table 10-5 lists the release history for this feature.

Table 10-5 Feature History for SNMP

Feature Name	Releases	Feature Information
IPv6 support	4.2(1)	Supports configuring IPv6 SNMP hosts.
Filter SNMP requests by community using an ACL	4.2(1)	Assigns an ACL to an SNMP community to filter SNMP requests. See the "Filtering SNMP Requests" section on page 10-11
Use interfaces for SNMP notification receivers	4.2(1)	Adds support to designate an interface to act as the source interface for SNMP notifications. See the "Configuring SNMP Notification Receivers" section on page 10-11
SNMP AAA synchronization	4.0(3)	Adds ability to modify the synchronized user configuration timeout. See the "Modifying the AAA Synchronization Time" section on page 10-23.
SNMP protocol	4.0(3)	Added ability to disable the SNMP protocol. See the "Disabling SNMP" section on page 10-23.

CHAPTER 11

Configuring RMON

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

This chapter includes the following sections:

Information About RMON

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

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

RMON is disabled by default and no events or alarms are configured in Cisco NX-OS. You can configure your RMON alarms and events by using the CLI or an SNMP-compatible network management station.

This section includes the following topics:

RMON Alarms

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

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

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

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

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

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

RMON Events

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

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

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

High Availability

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

Virtualization Support

Cisco NX-OS supports one instance of the RMON per virtual device context (VDC). By default, Cisco NX-OS places you in the default VDC. See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x.*

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

Licensing Requirements for RMON

Product	License Requirement
NX-OS	RMON requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme. For more information, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for RMON

RMON has the following prerequisites:

If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*, *Release 4.x*).

Guidelines and Limitations

RMON has the following configuration guidelines and limitations:

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

Configuring RMON

This section includes the following topics:

- Configuring RMON Alarms, page 11-4
- Configuring RMON Events, page 11-5

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

Configuring RMON Alarms

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

You can optionally specify the following parameters:

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

BEFORE YOU BEGIN

Ensure that you have configured an SNMP user and enabled SNMP notifications (see the "Configuring SNMP" section on page 10-7).

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

SUMMARY STEPS

1. **config t**
2. **rmon alarm** *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-index*] f**alling-threshold** *value* [*event-index*] [**owner** *name*]

   or

   rmon hcalarm *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold-high** *value* **rising-threshold-low** *value* [*event-index*] **falling-threshold-high** *value* **falling-threshold-low** *value* [*event-index*] [**owner** *name*] [**storagetype** *type*]
3. **show rmon** [**alarms** | **hcalarms**]
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.

	Command	Purpose
Step 2	**rmon alarm** *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-index*] **falling-threshold** *value* [*event-index*] [**owner** *name*] **Example:** switch(config)# rmon alarm 20 1.3.6.1.2.1.2.2.1.14.1 2900 delta rising-threshold 1500 1 falling-threshold 0 owner test	Creates an RMON alarm. The value range is from –2147483647 to 2147483647. The owner name can be any alphanumeric string.
	rmon hcalarm *index mib-object sample-interval* {**absolute** | **delta**} **rising-threshold-high** *value* **rising-threshold-low** *value* [*event-index*] **falling-threshold-high** *value* **falling-threshold-low** *value* [*event-index*] [**owner** *name*] [**storagetype** type] **Example:** switch(config)# rmon alarm 20 1.3.6.1.2.1.2.2.1.14.16777216 2900 delta rising-threshold-high 15 rising-threshold-low 151 falling-threshold-high 0 falling-threshold-low 0 owner test	Creates an RMON high capacity alarm. The value range is from –2147483647 to 2147483647. The owner name can be any alphanumeric string. The storage type range is from 1 to 5.
Step 3	**show rmon** {**alarms** | **hcalarms**} **Example:** switch(config)# show rmon alarms	(Optional) Displays information about rmon alarms or high capacity alarms.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring RMON Events

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

BEFORE YOU BEGIN

Ensure that you have configured an SNMP user and enabled SNMP notifications (see the "Configuring SNMP" section on page 10-7).

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

SUMMARY STEPS

1. **config t**
2. **rmon event** *index* [**description** *string*] [**log**] [**trap**] [**owner** *name*]

3. **show rmon events**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**rmon event** *index* [**description** *string*] [**log**] [**trap**] [**owner** *name*] **Example:** switch(config)# rmon event 1 trap	Configures an RMON event. The description string and owner name can be any alphanumeric string.
Step 3	**show rmon events** **Example:** switch(config)# show rmon events	(Optional) Displays information about rmon events.
Step 4	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Verifying RMON Configuration

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

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

RMON Example Configuration

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

```
config t
  rmon alarm 20 1.3.6.1.2.1.2.2.1.14.1 2900 delta rising-threshold 1500 1
falling-threshold 0 owner test
  rmon event 1 trap
```

Related Topics

See the following related topics:

- Configuring SNMP, page 10-1.

Default Settings

Table 11-1 lists the default settings for RMON parameters.

Table 11-1 Default RMON Parameters

Parameters	Default
Alarms	None configured.
Events	None configured.

Additional References

For additional information related to implementing RMON, see the following sections:

- Related Documents, page 11-7
- Standards, page 11-7
- MIBs, page 11-8

Related Documents

Related Topic	Document Title
RMON CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• RMON-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

CHAPTER 12

Configuring Online Diagnostics

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

This chapter includes the following sections:

Note For complete syntax and usage information for the commands in this chapter, see the *Cisco Nexus 7000 Series NX-OS System Management Command Reference*.

Information About Online Diagnostics

Online diagnostics help you verify that hardware and internal data paths are operating as designed so that you can rapidly isolate faults.

This section includes the following topics:

Online Diagnostic Overview

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. Table 12-1 describes the bootup diagnostic tests for a supervisor.

Table 12-1 Bootup Diagnostics

Test ID	Diagnostic	Description
1	ManagementPortLoopback	Disruptive test, not an on-demand test Tests loopback on the management port of a module.
2	EOBCPortLoopback	Disruptive test, not an on-demand test Ethernet out of band
4	USB	Nondisruptive test Checks the USB controller initialization on a module.
5	CryptoDevice	Nondisruptive test Checks the Cisco Trusted Security (CTS) device initialization on a module.

Note Modules run the EOBCPortLoopback diagnostic as a nondisruptive bootup test, using test ID 1.

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 Cisco NX-OS to either bypass the bootup diagnostics or run the complete set of bootup diagnostics. See the "Setting the Bootup Diagnostic Level" section on page 12-5.

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. Table 12-2 describes the health monitoring diagnostics and test IDs for a supervisor.

Table 12-2 Health Monitoring Nondisruptive Diagnostics for a Supervisor

Test ID	Diagnostic	Default Interval	Default Setting	Description
3	ASICRegisterCheck	20 seconds	active	Checks read/write access to scratch registers for the ASICs on a module.
5	PortLoopback	15 minutes	active	Verifies connectivity through every port that is administratively down on every module in the system.
6	NVRAM	30 seconds	active	Verifies the sanity of the NVRAM blocks on a supervisor.
7	RealTimeClock	5 minutes	active	Verifies that the real-time clock on the supervisor is ticking.
8	PrimaryBootROM	30 minutes	active	Verifies the integrity of the primary boot device on the supervisor.
9	SecondaryBootROM	30 minutes	active	Verifies the integrity of the secondary boot device on the supervisor.
10	CompactFlash	30 minutes	active	Verifies access to the internal compact flash devices.
11	ExternalCompactFlash	30 minutes	active	Verifies access to the external compact flash devices.
12	PwrMgmtBus	30 seconds	active	Verifies the standby power management control bus.
13	SpineControlBus	30 seconds	active	Verifies the availability of the standby spine module control bus.
14	SystemMgmtBus	30 seconds	active	Verifies the availability of the standby system management bus.
15	StatusBus	30 seconds	active	Verifies the status trasmitted the status bus for the supervisor, modules, and fabric cards.
16	StandbyFabricLoopback	60 seconds	active	Verifies the connectivity of the standby supervisor to the crossbars on the spine card.

Table 12-3 describes the health monitoring diagnostics for a module.

Table 12-3 Health Monitoring Nondisruptive Diagnostics for a Module

Test ID	Diagnostic	Default Interval	Default Setting	Description
2	ASICRegisterCheck	1 minute	active	Checks read/write access to scratch registers for the ASICs on a module.
3	PrimaryBootROM	30 minutes	active	Verifies the integrity of the primary boot device on a module.
4	SecondaryBootROM	30 minutes	active	Verifies the integrity of the secondary boot device on a module.
5	PortLoopback[1]	15 minutes	active	Tests the packet path from the supervisor module to the physical port in ADMIN DOWN state on modules.
6	RewriteEngineLoopback	60 seconds	active	Tests nondisruptive loopback for all ports up to the Rewrite Engine ASIC device.

[1] PortLoopback test supported on 32-port 10-Gbps Ethernet module and 48-port 1-G optical ethernet module.

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. See the "Starting or Stopping an On-Demand Diagnostic Test" section on page 12-8 for more information.

You can also modify the default interval for a health monitoring test. See the "Activating a Diagnostic Test" section on page 12-6 for more information.

High Availability

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

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

Virtualization Support

Cisco NX-OS supports online diagnostics in the default virtual device context (VDC). By default, Cisco NX-OS places you in the default VDC. See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x* for more information.

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
NX-OS	Online diagnostics require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme. For more information, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for Online Diagnostics

Online diagnostics have the following prerequisite:

- If you configure VDCs, install the Advanced Services license and go to the VDC that you want to configure. For more information, see the document, *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Guidelines and Limitations

You cannot run disruptive online diagnostic tests on demand.

Configuring Online Diagnostics

This section includes the following topics:

- Setting the Bootup Diagnostic Level, page 12-5
- Activating a Diagnostic Test, page 12-6
- Setting a Diagnostic Test as Inactive, page 12-8
- Starting or Stopping an On-Demand Diagnostic Test, page 12-8
- Clearing Diagnostic Results, page 12-9
- Simulating Diagnostic Results, page 12-10

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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **diagnostic bootup level** [*complete* | *bypass*]
3. **show diagnostic bootup level**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`config t`
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **diagnostic monitor interval module** *slot* **test** [*test-id* | *name* | **all**] **hour** *hour* **min** *minutes* **second** *sec*
3. **diagnostic monitor module** *slot* **test** [*test-id* | *name* | **all**]
4. **show diagnostic content module** {*slot* | **all**}

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**diagnostic monitor interval module** *slot* **test** [*test-id* | *name* | **all**] **hour** *hour* **min** *minutes* **second** *sec* **Example:** `switch(config)# diagnostic monitor interval module 6 test 3 hour 1 min 0 sec 0`	(Optional) Configures the interval at which the specified test is run. If no interval is set, the test runs at the interval set previously, or the default interval. The argument ranges are as follows: • slot—The range is from 1 to 10. • test-id—The range is from 1 to 14. • name—Can be any case-sensitive alphanumeric string up to 32 characters. • hour —The range is from 0 to 23 hours. • minute—The range is from 0 to 59 minutes. • second —The range is from 0 to 59 seconds.
Step 3	**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.
Step 4	**show diagnostic content module** {*slot* | **all**} **Example:** `switch(config)# show diagnostic content module 6`	(Optional) Displays information about the diagnostics and their attributes.

Setting a Diagnostic Test as Inactive

You can set a diagnostic test as inactive. Inactive tests keep their current configuration but do not run at at the scheduled interval.

Use the following command in global configuration mode to set a diagnostic test as inactive:

Command	Purpose
no diagnostic monitor module *slot* **test** [*test-id* | *name* | **all**] **Example:** `switch(config)# no diagnostic monitor interval module 6 test 3`	Inactivates 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.

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.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **diagnostic ondemand iteration** *number*
2. **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. **show diagnostic status module** *slot*

DETAILED STEPS

	Command	Purpose
Step 1	**diagnostic ondemand iteration** *number* **Example:** `switch# diagnostic ondemand iteration 5`	(Optional) Configures the number of times that the on-demand test runs. The range is from 1 to 999. The default is 1.

Step 2	**diagnostic ondemand action-on-failure** {**continue failure-count** *num-fails* | **stop**}
	Example:
	`switch# diagnostic ondemand action-on-failure stop`
		(Optional) 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	**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	**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	**show diagnostic status module** *slot* **Example:** `switch# show diagnostic status module 6`	(Optional) Verifies that the diagnostic has been scheduled.

Clearing Diagnostic Results

You can clear diagnostic test results.

Use the following command in any mode to clear the diagnostic test results:

Command	Purpose
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.

Simulating Diagnostic Results

You can simulate a diagnostic test result.

Use the following command in any mode to simulate a diagnostic test result:

Command	Purpose
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.

Use the following command in any mode to clear the simulated diagnostic test result:

Command	Purpose
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 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 results 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** | **fabric-utilization** | **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.

Online Diagnostic Example Configuration

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:

```
conf t
 diagnostic monitor module 6 test 2
 diagnostic monitor interval module 6 test 2 hour 3 min 30 sec 0
```

Default Settings

Table 12-4 lists the default settings for online diagnostic parameters.

Table 12-4 Default Online Diagnostic Parameters

Parameters	Default
Bootup diagnostics level	complete
Nondisruptive tests	active

Additional References

For additional information related to implementing online diagnostics, see the following sections:

- Related Documents, page 12-11
- Standards, page 12-12

Related Documents

Related Topic	Document Title
Online diagnostics CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for Online Diagnostics

Table 12-5 lists the release history for this feature.

Table 12-5 Feature History for Online Diagnostics

Feature Name	Releases	Feature Information
Updated GOLD tests	4.2(1)	Added support for PortLoopback, StatusBus, and StandbyFabricLoopback tests.
Online diagnostics (GOLD)	4.0(q)	Feature was introduced.

CHAPTER 13

Configuring the Embedded Event Manager

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

This chapter includes the following sections:

Information About EEM

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

This section includes the following topics:

EEM Overview

EEM consists of three major components:

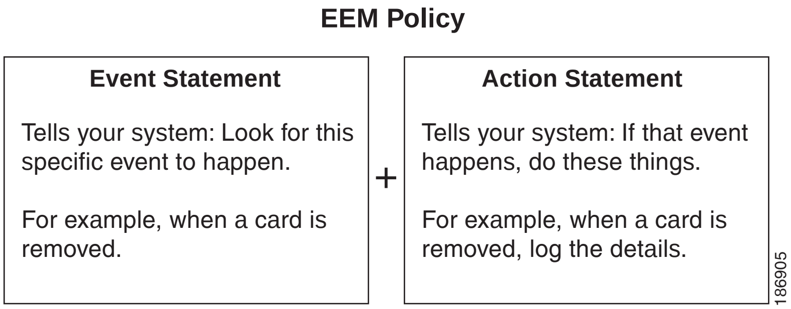
- Event statements—Events to monitor from another Cisco NX-OS component that may require some action, workaround, or notification.
- Action statements —An action that EEM can take, such as sending an e-mail, or disabling an interface, to recover from an event.
- Policies—An event paired with one or more actions to troubleshoot or recover from the event.

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.

Figure 13-1 shows the two basic statements in an EEM policy.

Figure 13-1 EEM Policy Statements


EEM Policy
Event Statement
Tells your system: Look for this specific event to happen.
For example, when a card is removed.
+
Action Statement
Tells your system: If that event happens, do these things.
For example, when a card is removed, log the details.
186905

You can configure EEM policies using the CLI or using 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 related to the same event as your policy. To configure a user policy, see the "Defining a User Policy Using the CLI" section on page 13-7.

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

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

To configure an overriding policy, see the "Overriding a Policy" section on page 13-14.

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.

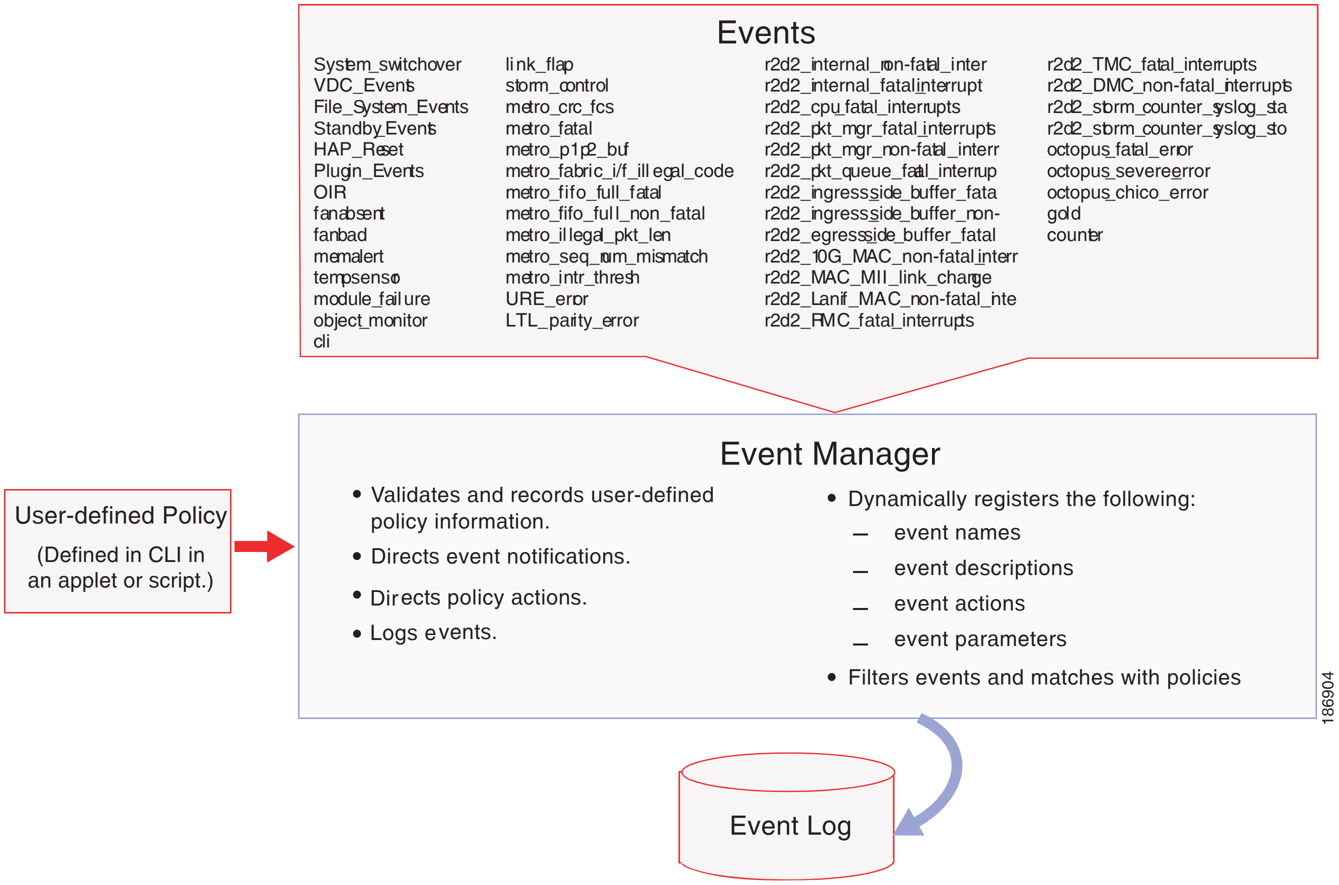
Event Statements

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.

Figure 13-2 shows events that are handled by EEM.

Figure 13-2 EEM Overview


Events
System_switchover
VDC_Events
File_System_Events
Standby Events
HAP_Reset
Plugin_Events
OIR
fanabsent
fanbad
memalert
tempsensor
module_failure
object_monitor
cli
link_flap
storm_control
metro_crc_fcs
metro_fatal
metro_p1p2_buf
metro_fabric_i/f_illegal_code
metro_fifo_full_fatal
metro_fifo_full_non_fatal
metro_illegal_pkt_len
metro_seq_num_mismatch
metro_intr_thresh
URE_error
LTL_parity_error
r2d2_internal_non-fatal_inter
r2d2_internal_fatal interrupt
r2d2_cpu fatal_interrupts
r2d2_pkt_mgr_fatal_interrupts
r2d2_pkt_mgr_non-fatal_interr
r2d2_pkt_queue_fatal_interrup
r2d2_ingress side_buffer_fata
r2d2_ingress side_buffer_non-
r2d2_egress side_buffer_fatal
r2d2_10G_MAC_non-fatal interr
r2d2_MAC_MII_link_change
r2d2_Lanif_MAC_non-fatal_inte
r2d2_RMC_fatal_interrupts
r2d2_TMC_fatal_interrupts
r2d2_DMC_non-fatal_interrupts
r2d2_storm_counter_syslog_sta
r2d2_storm_counter_syslog_sto
octopus fatal_error
octopus severe error
octopus chico_error
gold
counter
User-defined Policy
(Defined in CLI in an applet or script.)
Event Manager
• Validates and records user-defined policy information.
• Directs event notifications.
• Directs policy actions.
• Logs events.
• Dynamically registers the following:
– event names
– event descriptions
– event actions
– event parameters
• Filters events and matches with policies
Event Log
186904

Event statements specify the event that triggers a policy to run. You can configure only one event statement per policy.

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 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 VHS 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 script policy, copy it to the device and activate it. To configure a policy in a script, see the "Defining a Policy using a VSH Script" section on page 13-12.

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 e-mail server.

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

Example 13-1 shows a sample action statement to force a module 1 shutdown, with a reset reason of "EEM action."

Example 13-1 Action Statement

```
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 Example 13-2.

Example 13-2 Action Statement with Environment Variable

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

You can reuse this environment variable in any policy. For more information on environment variables, see the "Defining an Environment Variable" section on page 13-6.

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

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

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

See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*, for more information on VDCs.

Licensing Requirements for EEM

Product	License Requirement
NX-OS	EEM requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme. For more information, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for EEM

EEM has the following prerequisites:

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

Guidelines and Limitations

EEM has the following configuration guidelines and limitations:

- The maximum number of configurable EEM policies is 500.
- Action statements within your user policy or overriding policy should not negate each other or adversely affect the associated system policy.
- If you want to allow a triggered event to process any default actions, you must configure the EEM policy to allow the default action. For example, if you match a CLI command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the CLI command to execute.
- An override policy that consists of an event statement and no action statement triggers no action and no notification of failures.
- An override policy without an event statement overrides all possible events in the system policy.

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. For more information about system policies, see the “Embedded Event Manager System Events and Configuration Examples” appendix.

This section includes the following topics:

- Defining an Environment Variable, page 13-6
- Defining a User Policy Using the CLI, page 13-7
- Defining a Policy using a VSH Script, page 13-12
- Registering and Activating a VSH Script Policy, page 13-13
- Overriding a Policy, page 13-14
- Configuring Memory Thresholds, page 13-15

Defining an Environment Variable

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **event manager environment** *variable-name variable-value*
3. **show event manager environment**
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in 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	**`show event manager environment`**
	Example:
	`switch(config-applet)# show event manager environment`
		(Optional) Displays information about the configured environment variables.

Step 4	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Defining a User Policy Using the CLI

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

This section includes the following topics:

- Configuring Event Statements, page 13-8
- Configuring Action Statements, page 13-11

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **event manager applet** *applet-name*
3. **description** *policy-description*

4. **event** *event-statement*
5. **action** *number*[**.***number2*] *action-statement*
   (Repeat Step 5 for multiple action statements.)
6. **show event manager policy-state** *name* [**module** *module-id*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in 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	**description** *policy-description* **Example:** `switch(config-applet)# description "Monitors interface shutdown."`	(Optional) Configures a descriptive string for the policy. The string can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks.
Step 4	**event** *event-statement* **Example:** `switch(config-applet)# event cli match "shutdown"`	Configures the event statement for the policy. See the "Configuring Event Statements" section on page 13-8.
Step 5	**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. See the "Configuring Action Statements" section on page 13-11. Repeat Step 5 for multiple action statements.
Step 6	**show event manager policy-state** *name* [**module** *module-id*] **Example:** `switch(config-applet)# show event manager policy-state monitorShutdown`	(Optional) Displays information about the status of the configured policy.
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

Configuring Event Statements

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

Command	Purpose
event cli match *expression* [**count** repeats | **time** *seconds*] **Example:** switch(config-applet)# event cli match “shutdown”	Triggers an event if you enter a command that matches the regular expression. The *repeats* range is from 0 to 65000. The time range, in seconds, is from 0 to 4294967295, where 0 indicates no time limit.
event counter 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 *counter* name can be any case-sensitive, alphanumeric string up to 28 characters. The *entry* and *exit* value ranges are from 0 to 2147483647.
event fanabsent [**fan** *number*] **time** *seconds* **Example:** switch(config-applet)# event fanabsent time 300	Triggers an event if a fan is removed from the device for more than the configured time, in seconds. The *number* range is module dependent. The *seconds* range is from 10 to 64000.
event fanbad [**fan** *number*] **time** *seconds* **Example:** switch(config-applet)# event fanbad time 3000	Triggers an event if a fan fails for more than the configured time, in seconds. The *number* range is module dependent. The *seconds* range is from 10 to 64000.
event 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 memory {**critical** | **minor** | **severe**} **Example:** switch(config-applet)# event memory critical	Triggers an event if a memory threshold is crossed. See also the “Configuring Memory Thresholds” section on page 13-15.
event module-failure 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. See the *Cisco Nexus 7000 Series NX-OS System Management Command Reference*, for information on the failure types. The *repeats* range is from 0 to 4294967295. The *seconds* range is from 0 to 4294967295, where 0 indicates no time limit.
event module 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.

Command	Purpose
event oir {**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. You can optionally configure a specific fan, module, or power supply number. The *number* range is as follows: • Fan number—Module dependent. • Module number—Device dependent. • Power supply number—The range is from 1 to 3.
event policy-default count *repeats* [**time** *seconds*] **Example:** `switch(config-applet)# event policy-default count 3`	Uses the event configured in the system policy. Use this option for overriding policies. The *repeats* range is from 1 to 65000. The *seconds* range is from 0 to 4294967295, where 0 indicates no time limit.
event poweroverbudget **Example:** `switch(config-applet)# event poweroverbudget`	Triggers an event if the power budget exceeds the capacity of the configured power supplies.
event snmp 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 *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 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 counet 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.

Command	Purpose

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.

event track *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 *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.

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.

Command	Purpose
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*] **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.
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

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

Ensure that you are logged in with administrator privileges.

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

DETAILED STEPS

Step 1 In a text editor, list the commands that define the policy.

Step 2 Name the text file and save it.

Step 3 Copy the file to the following system directory:

bootflash://eem/user_script_policies

Registering and Activating a VSH Script Policy

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **event manager policy** *policy-script*
3. **show event manager policy** *name*
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in 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	**`show event manager policy`** *`name`*
	Example:
	`switch(config-applet)# show event manager policy moduleScript`
		(Optional) Displays information about the configured policy.

Step 4	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Overriding a Policy

You can override a system policy.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **show event manager policy-state** *system-policy*
3. **event manager applet** *applet-name* **override** *system-policy*
4. **description** *policy-description*
5. **event** *event-statement*
6. **action** *number action-statement*
   (Repeat Step 6 for multiple action statements.)
7. **show event manager policy-state** *name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`show event manager policy-state`** *`system-policy`*
	
	Example:
	`switch(config-applet)# show event manager policy-state __ethpm_link_flap`
	`Policy __ethpm_link_flap`
	`  Cfg count : 5`
	`  Cfg time interval : 10.000000 (seconds)`
	`    Hash default, Count 0`
		(Optional) 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 the "Embedded Event Manager System Events and Configuration Examples" appendix.

Step 3	**`event manager applet`** *`applet-name`* **`override`** *`system-policy`*
	
	Example:
	`switch(config)# event manager applet ethport override __ethpm_link_flap`
	`switch(config-applet)#`
		Overrides a system policy and enters applet configuration mode. The *applet-name* can be any case-sensitive alphanumeric string up to 29 characters. The *system-policy* must be one of the existing system policies.

	Command	Purpose
Step 4	**description** *policy-description* **Example:** `switch(config-applet)# description "Overrides link flap policy."`	(Optional) Configures a descriptive string for the policy. The string can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks.
Step 5	**event** *event-statement* **Example:** `switch(config-applet)# event policy-default count 2 time 1000`	Configures the event statement for the policy. See the "Configuring Event Statements" section on page 13-8.
Step 6	**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. See the "Configuring Action Statements" section on page 13-11. Repeat Step 6 for multiple action statements.
Step 7	**show event manager policy-state** *name* **Example:** `switch(config-applet)# show event manager policy-state ethport`	(Optional) Displays information about the configured policy.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

Configuring Memory Thresholds

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

BEFORE YOU BEGIN

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

Ensure that you are logged in with administrator privileges.

SUMMARY STEPS

1. **config t**
2. **system memory-thresholds minor** *minor* **severe** *severe* **critical** *critical*
3. **system memory-thresholds threshold critical no-process-kill**
4. **show running-config | include "system memory"**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in 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:
		• 2009 May 7 17:06:30 switch %$ VDC-1 %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR
		• 2009 May 7 17:06:30 switch %$ VDC-1 %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE
		• 2009 May 7 17:06:30 switch %$ VDC-1 %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL
		• 2009 May 7 17:06:35 switch %$ VDC-1 %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR ALERT RECOVERED
		• 2009 May 7 17:06:35 switch %$ VDC-1 %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE ALERT RECOVERED
		• 2009 May 7 17:06:35 switch %$ VDC-1 %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL ALERT RECOVERED

Step 3	**`system memory-thresholds threshold critical no-process-kill`**
	Example:
	`switch(config)# system memory-thresholds threshold critical no-process-kill`
		(Optional) 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.

	Command	Purpose
Step 4	**`show running-config	include “system memory”`** **Example:** `switch(config-applet)# show running-config	include “system memory”`	(Optional) Displays information about the system memory configuration.

Step 5	**`copy running-config startup-config`**
	Example:
	`switch(config)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Verifying 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 internal [*policy-name*] [**inactive**]	Displays information about the configured policies.
show event manager policy-state *policy-name*	Displays information about the policy state, including thresholds.
show event manager script system [*policy-name* | **all**]	Displays information about the script policies.
show event manager system-policy [**all**]	Displays information about the predefined system policies.
show running-config eem	Displays information about the running configuration for EEM.
show startup-config eem	Displays information about the startup configuration for EEM.

EEM Example Configuration

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 "Confiiguration change"
  action 2.0 event-default
```

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

Default Settings

Table 13-1 lists the default settings for EEM parameters.

Table 13-1 Default EEM Parameters

Parameters	Default
system policies	Active.

Additional References

For additional information related to implementing EEM, see the following sections:

- Related Documents, page 13-19
- Standards, page 13-19

Related Documents

Related Topic	Document Title
EEM commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for EEM

Table 13-2 lists the release history for this feature.

Table 13-2 Feature History for EEM

Feature Name	Releases	Feature Information
Memory thresholds	4.2(4)	Changed the minor, severe, and critical default memory thresholds from 70, 80, and 90 to 85, 90, and 95.
Memory Thresholds Configuration	4.1(3)	Added a configuration section for memory thresholds. See the "Configuring Memory Thresholds" section on page 13-15.

CHAPTER 14

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:

Information 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 sessions on the local device.

This section includes the following topics:

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, egress, or both directions of traffic. SPAN sources include the following:

- Ethernet ports
- VLANs—When a VLAN is specified as a SPAN source, all supported interfaces in the VLAN are SPAN sources.
- Remote SPAN (RSPAN) VLANs
- The inband interface to the control plane CPU—You can monitor the inband interface only from the default VDC. Inband traffic from all VDCs is monitored.

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

Characteristics of Source Ports

SPAN source ports have the following characteristics:

- A port configured as a source port cannot also be configured as a destination port.
- An RSPAN VLAN can only be used as a SPAN source.
- If you use the supervisor inband interface as a SPAN source, the following packets are monitored:
  - All packets that arrive on the supervisor hardware (ingress)
  - All packets generated by the supervisor hardware (egress)

SPAN Destinations

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

Characteristics of Destination Ports

SPAN destination ports have the following characteristics:

- Destinations for a SPAN session include Ethernet ports or port-channel interfaces in either access or trunk mode.
- 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.
- An RSPAN VLAN cannot be used as a SPAN destination.
- You can configure SPAN destinations to inject packets to disrupt a certain TCP packet stream in support of the Intrusion Detection System (IDS).
- You can configure SPAN destinations to enable a forwarding engine to learn the MAC address of the IDS.

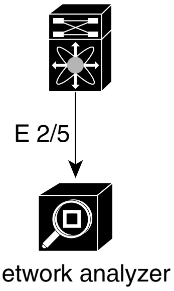
SPAN Sessions

You can create up to 18 SPAN sessions designating sources and destinations to monitor.

Note Only two SPAN sessions can be running simultaneously.

Figure 14-1 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 14-1 SPAN Configuration



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

186283

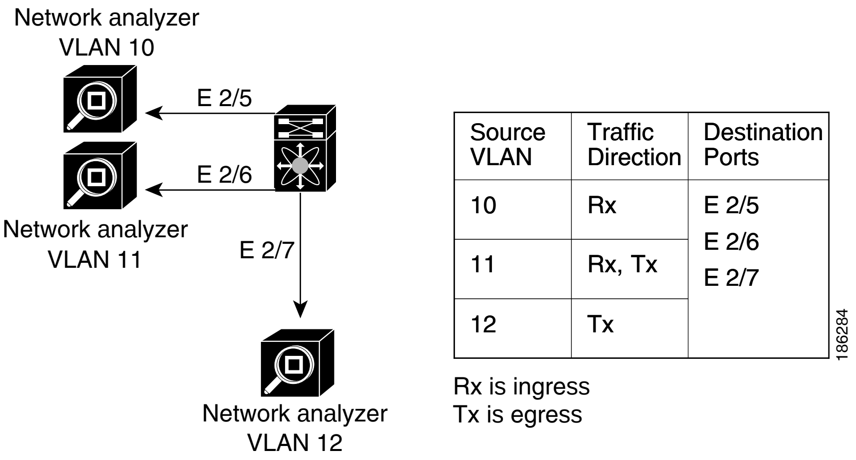
Virtual SPAN Sessions

You can create a virtual SPAN session to monitor multiple VLAN sources and choose only VLANs of interest to transmit on multiple destination ports. For example, you can configure SPAN on a trunk port and monitor traffic from different VLANs on different destination ports.

Figure 14-2 shows a virtual SPAN configuration. The virtual SPAN session copies traffic from the three VLANs to the three specified destination ports. You can choose which VLANs to allow on each destination port to limit the traffic that the device transmits on it. In Figure 14-2, the device transmits packets from one VLAN at each destination port.

Note Virtual SPAN sessions cause all source packets to be copied to all destinations, whether the packets are required at the destination or not. VLAN traffic filtering occurs at the egress destination port level.

Figure 14-2 Virtual SPAN Configuration



Source VLAN	Traffic Direction	Destination Ports
10	Rx	E 2/5 E 2/6 E 2/7
11	Rx, Tx	
12	Tx	

Rx is ingress
Tx is egress

For information about configuring a virtual SPAN session, see the "Configuring a Virtual SPAN Session" section on page 14-10.

Multiple SPAN Sessions

Although you can define up to 18 SPAN sessions, only two SPAN sessions can be running simultaneously. You can shut down an unused SPAN session.

For information about shutting down SPAN sessions, see the "Shutting Down or Resuming a SPAN Session" section on page 14-13.

High Availability

The SPAN feature supports stateless and stateful restarts. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration.

Virtualization Support

A virtual device context (VDC) is a logical representation of a set of system resources. SPAN applies only to the VDC where the commands are entered.

Note You can monitor the inband interface only from the default VDC. Inband traffic from all VDCs is monitored.

For information about configuring VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for SPAN

The following table shows the licensing requirements for this feature:

Product	License Requirement
NX-OS	SPAN requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme, see the *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 7000 Series NX-OS Interfaces Configuration Guide, Release 4.x*.

Guidelines and Limitations

SPAN has the following configuration guidelines and limitations:

- Table 1 lists the SPAN session limits.

Table 1 SPAN Session Limits

Description	Limit
Configured SPAN sessions	18
Simultaneously running SPAN sessions	2
Source interfaces per session	128
Source VLANs per session[1]	32
Destination interfaces per session	32

[1] If you specify a VLAN range greater than 32, the first 32 VLANs are added as source VLANs to the SPAN session even if the VLANs have not been created. For example, if you specify a VLAN range of 1-40 for the SPAN session, only VLANs 1-32 are added to the SPAN session. To add only specific VLANs within a range, you must add the VLANs explicitly.

- SPAN is not supported for management ports.
- A destination port can only be configured in one SPAN session at a time.
- You cannot configure a port as both a source and destination port.
- A single SPAN session can include mixed sources in any combination of the following:
  - Ethernet ports, but not subinterfaces.
  - VLANs, which can be assigned to port channel subinterfaces
  - The inband interface to the control plane CPU
- Destination ports do not participate in any spanning tree instance. SPAN output includes Bridge Protocol Data Unit (BPDU) Spanning-Tree Protocol hello packets.

- When a SPAN session contains source ports that are monitored in the transmit or transmit and receive direction, packets that these ports receive may 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 include:
  - Traffic that results from flooding
  - Broadcast and multicast traffic
- For VLAN SPAN sessions with both ingress and egress configured, two packets (one from ingress and one from egress) are forwarded from the destination port if the packets get switched on the same VLAN.
- VLAN SPAN monitors only the traffic that leaves or enters Layer 2 ports in the VLAN.
- You can monitor the inband interface only from the default VDC. Inband traffic from all VDCs is monitored.
- You can configure an RSPAN VLAN for use only as a SPAN session source.
- You can configure a SPAN session on the local device only.
- SPAN sessions cannot capture packets with broadcast or multicast MAC addresses that reach the supervisor, such as ARP requests and Open Shortest Path First (OSPF) protocol hello packets, if the source of the session is the supervisor ethernet in-band interface. To capture these packets, you must use the physical interface as the source in the SPAN sessions.
- All SPAN replication is performed in the hardware. The supervisor CPU is not involved.

Configuring SPAN

This section includes the following topics:

- Configuring a SPAN Session, page 14-6
- Configuring a Virtual SPAN Session, page 14-10
- Configuring an RSPAN VLAN, page 14-12
- Shutting Down or Resuming a SPAN Session, page 14-13

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

Configuring a SPAN Session

You can configure a SPAN session on the local device only. By default, SPAN sessions are created in the shut state.

For sources, you can specify Ethernet ports, port channels, the supervisor inband interface, VLANs, and RSPAN VLANs. You can specify private VLANs (primary, isolated, and community) in SPAN sources.

A single SPAN session can include mixed sources in any combination of Ethernet ports, VLANs, or the inband interface to the control plane CPU. You cannot specify Ethernet port subinterfaces as sources for a SPAN session.

Note To use a Layer 3 port-channel subinterface as a SPAN source in the monitor session, you must specify the vlan ID that you entered when configuring IEEE 802.1Q VLAN encapsulation for the subinterface as the filter VLAN. When you use the main interface and the SPAN VLAN filter to filter the 802.1Q VLANs on the subinterfaces, SPAN shows the traffic for all subinterfaces on the SPAN destination port.

When you specify the supervisor inband interface for a SPAN source, the device monitors all packets that arrive on the supervisor hardware (ingress) and all packets generated by the supervisor hardware (egress).

For destination ports, you can specify Ethernet ports or port-channels in either access or trunk mode. You must enable monitor mode on all destination ports.

BEFORE YOU BEGIN

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

- You must have already configured the destination ports in access or trunk mode. For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.x* at the following.l

SUMMARY STEPS

1. **config t**
2. **interface ethernet** *slot*/*port*[-*port*]
3. **switchport**
4. **switchport mode** [**access** | **trunk** | **private-vlan**]
5. **switchport monitor** [**ingress** [**learning**]]
6. Repeat Steps 2 and 3 to configure monitoring on additional SPAN destinations.
7. **no monitor session** *session-number*
8. **monitor session** *session-number*
9. **description** *description*
10. **source** {**interface** *type* | **vlan** {*number* | *range*} [**rx** | **tx** | **both**]
11. Repeat Step 8 to configure all SPAN sources.
12. **filter vlan** {*number* | *range*}
13. Repeat Step 10 to configure all source VLANs to filter.
14. **destination interface** *type* {*number* | *range*}
15. Repeat Step 12 to configure all SPAN destination ports.
16. **no shut**
17. **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
18. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`interface ethernet`** *`slot`*`/`*`port`*`[-`*`port`*`]`
	Example:
	`switch(config)# interface ethernet 2/5`
	`switch(config-if)#`
		Enters interface configuration mode on the selected slot and port or range of ports.

Step 3	**`switchport`**
	Example:
	`switch(config-if)# switchport`
	`switch(config-if)#`
		Configures switchport parameters for the selected slot and port or range of ports.

Step 4	**`switchport mode`** `[`**`access`** `|` **`trunk`** `|` **`private-vlan`**`]`
	Example:
	`switch(config-if)# switchport mode trunk`
	`switch(config-if)#`
		Configures the switchport mode for the selected slot and port or range of ports.
		• access
		• trunk
		• private-vlan

Step 5	**`switchport monitor`** `[`**`ingress`** `[`**`learning`**`]]`
	Example:
	`switch(config-if)# switchport monitor`
		Configures the switchport interface as a SPAN destination:
		• **ingress**
		Allows the SPAN destination port to inject packets that disrupt a certain TCP packet stream, for example, in networks with IDS.
		• **ingress learning**
		Allows the SPAN destination port to inject packets, and learning adds the ability for the switch to learn the MAC address of the downstream network analysis device.

Step 6	(Optional) Repeat Steps 2 and 3 to configure monitoring on additional SPAN destinations.	—

Step 7	**`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 8	**`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. By default, the session is created in the shut state.

Step 9	**`description`** *`description`*
	Example:
	`switch(config-monitor)# description my_span_session_3`
		Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

	Command	Purpose

Step 10	**source** {**interface** *type* | **vlan** {*1-3967,4048-4093*}} [**rx** | **tx** | **both**]
	Example 1:
	`switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx`
	Example 2:
	`switch(config-monitor)# source interface port-channel 2`
	Example 3:
	`switch(config-monitor)# source interface sup-eth 0 both`
	Example 4:
	`switch(config-monitor)# source vlan 3, 6-8 tx`
		Configures sources and the traffic direction in which to copy packets. You can enter a range of Ethernet ports, a port channel, an inband interface, or a range of VLANs.
		You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify up to 128 interfaces. The VLAN range is from 1 to 3967 and 4048 to 4093.
		You can specify the traffic direction to copy as ingress (rx), egress (tx), or both. By default, the direction is both.
		Note You can monitor the inband interface only from the default VDC. The inband traffic from all VDCs is monitored.

Step 11	(Optional) Repeat Step 8 to configure all SPAN sources.	—

Step 12	**filter vlan** {*number* | *range*}
	Example:
	`switch(config-monitor)# filter vlan 3-5, 7`
		Configures which VLANs to select from the configured sources. You can configure one or more VLANs, as either a series of comma-separated entries, or a range of numbers. The VLAN range is from 1 to 3967 and 4048 to 4093.

Step 13	(Optional) Repeat Step 10 to configure all source VLANs to filter.	—

Step 14	**destination interface** *type* {*number* | *range*}
	Example:
	`switch(config-monitor)# destination interface ethernet 2/5, ethernet 3/7`
		Configures destinations for copied source packets. You can configure one or more destinations, as either a series of comma-separated entries, or a range of numbers. You can specify up to 128 interfaces.
		Note SPAN destination ports must be either access or trunk ports.

Step 15	(Optional) Repeat Step 12 to configure all SPAN destination ports.	—
Step 16	**no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state. **Note** Only two SPAN sessions can be running simultaneously.
Step 17	**show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-monitor)# show monitor session 3`	(Optional) Displays the SPAN configuration.
Step 18	**copy running-config startup-config** **Example:** `switch(config-monitor)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Virtual SPAN Session

You can configure a virtual SPAN session to copy packets from source ports, VLANs, and RSPAN VLANs to destination ports on the local device. By default, SPAN sessions are created in the shut state.

For sources, you can specify ports, VLANs, or RSPAN VLANs.

For destination ports, you can specify Ethernet ports. You can choose which VLANs to allow on each destination port to limit the traffic that the device transmits on it.

BEFORE YOU BEGIN

- Ensure that you are in the correct VDC (or use the **switchto vdc** command).
- You have already configured the destination ports in trunk mode. For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.x*.
- You have already configured the destination ports to monitor a SPAN session with the **switchport monitor** command.

SUMMARY STEPS

1. **config t**
2. **no monitor session** *session-number*
3. **monitor session** *session-number*
4. **source** {**interface** *type* | **vlan**} {*number* | *range*} [**rx** | **tx** | **both**]
5. Repeat Step 4 to configure all virtual SPAN VLAN sources.
6. **destination interface** *type* {*number* | *range*}
7. Repeat Step 6 to configure all virtual SPAN destination ports.
8. **no shut**
9. **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
10. **interface ethernet** *slot*/*port*[-*port*]
11. **switchport**
12. **switchport mode trunk**
13. **switchport trunk allowed vlan** {{*number* | *range*} | **add** {*number* | *range*} | **except** {*number* | *range*} | **remove** {*number* | *range*} | **all** | **none**}
14. Repeat Steps 10 and 11 to configure the allowed VLANs on each destination port.
15. **show interface ethernet** *slot*/*port*[-*port*] **trunk**
16. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**no monitor session** *session-number* **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified SPAN session. New session configuration is added to the existing session configuration.

Step 3	**monitor session** *session-number*
	Example:
	`switch(config)# monitor session 3`
	`switch(config-monitor)#`
		Enters the monitor configuration mode. A new session configuration is added to the existing session configuration.

Step 4	**source** {**interface** *type* | **vlan**} {*number* | *range*} [**rx** | **tx** | **both**] **Example:** `switch(config-monitor)# source vlan 3, 6-8 tx`	Configures sources and the traffic direction in which to copy packets. You can configure one or more sources, as either a series of comma-separated entries, or a range of numbers. You can specify up to 128 interfaces. The VLAN range is from 1 to 3967 and 4048 to 4093. You can specify the traffic direction to copy as ingress (rx), egress (tx), or both. By default, the direction is both.
Step 5	(Optional) Repeat Step 4 to configure all virtual SPAN source VLANs.	—

Step 6	**destination interface** *type* {*number* | *range*}
	Example:
	`switch(config-monitor)# destination interface ethernet 2/5, ethernet 3/7`
		Configures destinations for copied source packets. You can configure one or more interfaces, as either a series of comma-separated entries, or a range of numbers. The allowable range is from 1 to 128.
		Note Configure destination ports as trunk ports. For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.x*.

Step 7	(Optional) Repeat Step 6 to configure all virtual SPAN destination ports.	—
Step 8	**no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state. **Note** Only two SPAN sessions can be running simultaneously.
Step 9	**show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-monitor)# show monitor session 3`	(Optional) Displays the virtual SPAN configuration.

	Command	Purpose
Step 10	**interface ethernet** *slot*/*port*[-*port*] **Example:** switch(config)# interface ethernet 2/5 switch(config-if)#	Enters interface configuration mode on the selected slot and port or range of ports.
Step 11	**switchport** **Example:** switch(config-if)# switchport switch(config-if)#	Makes interface a Layer 2 interface.
Step 12	**switchport mode trunk** **Example:** switch(config-if)# switchport mode trunk switch(config-if)#	Puts Layer 2 interface into trunk mode.
Step 13	**switchport trunk allowed vlan** {{*number* | *range*} | **add** {*number* | *range*} | **except** {*number* | *range*} | **remove** {*number* | *range*} | **all** | **none**} **Example:** switch(config-if)# switchport trunk allowed vlan 3-5	Configures the range of VLANS that are allowed on the interface. You can add to or remove from the existing VLANs, you can select all VLANs except those VLANs that you specify, or you can select all or none of the VLANs. By default, all VLANs are allowed on the interface. You can configure one or more VLANs, as either a series of comma-separated entries, or a range of numbers. The VLAN range is from 1 to 3967 and 4048 to 4093.
Step 14	(Optional) Repeat Steps 10 and 11 to configure the allowed VLANs on each destination port.	—
Step 15	**show interface ethernet** *slot*/*port*[-*port*] **trunk** **Example:** switch(config-if)# show interface ethernet 2/5 trunk	(Optional) Displays the interface trunking configuration for the selected slot and port or range of ports.
Step 16	**copy running-config startup-config** **Example:** switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Configuring an RSPAN VLAN

You can specify a remote SPAN (RSPAN) VLAN as a SPAN session source.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **vlan** *vlan*

3. **remote-span**
4. **exit**
5. **show vlan**
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**`vlan`** *`vlan`*
	
	Example:
	`switch(config)# vlan 901`
	`switch(config-vlan)#`
		Enters VLAN configuration mode for the VLAN specified.

Step 3	**`remote-span`**
	
	Example:
	`switch(config-vlan)# remote-span`
		Configures the VLAN as an RSPAN VLAN.

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

Step 5	**`show vlan`**
	
	Example:
	`switch(config)# show vlan`
		(Optional) Displays the VLAN configuration. Remote SPAN VLANs are listed together.

Step 6	**`copy running-config startup-config`**
	
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		(Optional) 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. Because only two SPAN sessions can be running simultaneously, 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.

BEFORE YOU BEGIN

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **config t**
2. **monitor session** {*session-range* | **all**} **shut**
3. **no monitor session** {*session-range* | **all**} **shut**
4. **monitor session** *session-number*
5. **shut**
6. **no shut**
7. **show monitor**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`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 SPAN sessions. The session ranges from 1 to 18. By default, sessions are created in the shut state. Only two sessions can be running at a time.

Step 3	**`no monitor session`** {*`session-range`* | **`all`**} **`shut`**
	Example:
	`switch(config)# no monitor session 3 shut`
		Resumes (enables) the specified SPAN sessions. The session ranges from 1 to 18. By default, sessions are created in the shut state. Only two sessions can be running at a time.
		Note 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`*
	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 5	**`shut`**
	Example:
	`switch(config-monitor)# shut`
		Shuts down the SPAN session. By default, the session is created in the shut state.

	Command	Purpose
Step 6	**no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state. **Note** Only two SPAN sessions can be running simultaneously.
Step 7	**show monitor** **Example:** `switch(config-monitor)# show monitor`	(Optional) Displays the status of SPAN sessions.
Step 8	**copy running-config startup-config** **Example:** `switch(config-monitor)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying the SPAN Configuration

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

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the SPAN session configuration.

For detailed information about the fields in the output from these commands, see the *Cisco NX-OS System Management Command Reference*.

SPAN Example Configurations

This section includes the following topics:

- SPAN Session Example Configuration, page 14-15
- Virtual SPAN Session Example Configuration, page 14-16
- Private VLAN Source in SPAN Session Example Configuration, page 14-16

SPAN Session Example Configuration

To configure a SPAN session, follow these steps:

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

```
switch# config t
  switch(config)# interface ethernet 2/5
    switch(config-if)# switchport
    switch(config-if)# switchport mode trunk
    switch(config-if)# switchport monitor
    switch(config-if)# no shut
    switch(config-if)# exit
  switch(config)#
```

Step 2 Configure a SPAN session.

```
  switch(config)# no monitor session 3
  switch(config)# monitor session 3
    switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
    switch(config-monitor)# source interface port-channel 2
    switch(config-monitor)# source interface sup-eth 0 both
    switch(config-monitor)# source vlan 3, 6-8 tx
    switch(config-monitor)# filter vlan 3-5, 7
    switch(config-monitor)# destination interface ethernet 2/5
    switch(config-monitor)# no shut
    switch(config-monitor)# exit
  switch(config)# show monitor session 3
  switch(config)# copy running-config startup-config
```

Virtual SPAN Session Example Configuration

To configure a virtual SPAN session, follow these steps:

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

```
switch# config t
  switch(config)# interface ethernet 3/1
    switch(config-if)# switchport
    switch(config-if)# switchport mode trunk
    switch(config-if)# switchport trunk allowed vlan add 100-200
    switch(config-if)# switchport monitor
    switch(config-if)# no shut
    switch(config-if)# exit
  switch(config)# interface ethernet 3/2
    switch(config-if)# switchport
    switch(config-if)# switchport mode trunk
    switch(config-if)# switchport trunk allowed vlan add 201-300
    switch(config-if)# switchport monitor
    switch(config-if)# no shut
    switch(config-if)# exit
  switch(config)#
```

Step 2 Configure a SPAN session.

```
  switch(config)# no monitor session 3
  switch(config)# monitor session 3
    switch(config-monitor)# source vlan 100-300
    switch(config-monitor)# destination interface ethernet 3/1-2
    switch(config-monitor)# no shut
    switch(config-monitor)# exit
  switch(config)# show monitor session 3
  switch(config)# copy running-config startup-config
```

Private VLAN Source in SPAN Session Example Configuration

To configure a SPAN session that includes a private VLAN source, follow these steps:

Step 1 Configure source VLANs.

```
switch# config t
  switch(config)# vlan 100
    switch(config-vlan)# private-vlan primary
    switch(config-vlan)# exit
  switch(config)# interface ethernet 3/1
    switch(config-if)# switchport
    switch(config-if)# switchport access vlan 100
    switch(config-if)# no shut
    switch(config-if)# exit
  switch(config)# interface ethernet 3/2
    switch(config-if)# switchport
    switch(config-if)# switchport mode trunk
    switch(config-if)# switchport trunk native vlan 100
    switch(config-if)# no shut
    switch(config-if)# exit
  switch(config)#
```

Step 2 Configure destination ports in access or trunk mode, and enable SPAN monitoring.

```
switch# config t
  switch(config)# interface ethernet 3/3
    switch(config-if)# switchport
    switch(config-if)# switchport mode trunk
    switch(config-if)# switchport trunk allowed vlan add 100-200
    switch(config-if)# switchport monitor
    switch(config-if)# no shut
    switch(config-if)# exit
  switch(config)#
```

Step 3 Configure a SPAN session.

```
  switch(config)# no monitor session 3
  switch(config)# monitor session 3
    switch(config-monitor)# source vlan 100
    switch(config-monitor)# destination interface ethernet 3/3
    switch(config-monitor)# no shut
    switch(config-monitor)# exit
  switch(config)# show monitor session 3
  switch(config)# copy running-config startup-config
```

Additional References

For additional information related to implementing SPAN, see the following sections:

- Related Documents, page 14-18
- Standards, page 14-18

Related Documents

Related Topic	Document Title
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*
SPAN commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for SPAN

Table 14-2 lists the release history for this feature.

Table 14-2 Feature History for SPAN

Feature Name	Releases	Feature Information
Guidelines and Limitations	4.1(3)	Added a table of SPAN session limits. See the "Table 1SPAN Session Limits" section on page 14-5.

CHAPTER 15

Configuring Onboard Failure Logging

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

This chapter includes the following sections:

Information About OBFL

This section includes the following topics:

OBFL Overview

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.

The data stored by OBFL include the following:

- 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

OBFL stores a kernel trace in case Cisco NX-OS crashes.

Virtualization Support

You must be in the default virtual device context (VDC) to configure and display OBFL information. See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x* for more information on VDCs.

Licensing Requirements for OBFL

The following table shows the licensing requirements for this feature:

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

Prerequisites for OBFL

If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x).*

You must have network-admin user privileges and be logged into the default VDC.

Guidelines and Limitations

OBFL has the following guidelines and limitations:

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

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

Configuring OBFL

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

BEFORE YOU BEGIN

Make sure you are in global configuration mode.

SUMMARY STEPS

1. **hw-module loggine onboard**
2. **hw-module logging onboard environmental-history**
3. **hw-module logging onboard error-stats**
4. **hw-module logging onboard interrupt-stats**
5. **hw-module logging onboard module** *slot*
6. **hw-module logging onboard module obfl-log**
7. **show logging onboard**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	`Example:`
	`switch# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 1	**`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.

	Command	Purpose

Step 2	**`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 3	**`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 4	**`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 5	**`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 6	**`hw-module logging onboard obfl-log`**
	
	`Example:`
	`switch(config)# hw-module logging onboard obfl-log`
	`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.

	Command	Purpose
Step 7	**show logging onboard** **Example:** **switch(config)# show logging onboard** ---------------------------- OBFL Status ---------------------------- Switch OBFL Log: Enabled Module: 7 OBFL Log: Enabled cpu-hog Enabled environmental-history Enabled error-stats Enabled exception-log . . . Fri Mar 21 19:07:33 2008 (957597 us) Module 2 SecondaryBootROM test has failed 20 times with error BIOS file checksum error Library could not be opened *** /lc/isan/lib/libcrdcfg.so: undefined symbol: get_slot_id *** plog_show_data_type: Error opening library statcl, func_name statcl_disp_func	(Optional) Displays information about OBFL.
Step 8	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Verifying the OBFL Configuration

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:  2 OBFL Log:                                Enabled
    cpu-hog                                             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
    stack-trace                                         Enabled
    system-health                                       Enabled

    Module:  6 OBFL Log:                                Enabled
    cpu-hog                                             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
    stack-trace                                         Enabled
    system-health                                       Enabled
    temp Error                                          Enabled
```

Use the following commands to display OBFL information stored in flash on a module:

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 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 kernel-trace	Displays kernel trace information.
show logging onboard module *slot*	Displays OBFL information for a specific module.
show logging onboard obfl-history	Displays history information.
show logging onboard obfl-logs	Displays log information.

Command	Purpose
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.

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

OBFL Example Configuration

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

```
conf t
 hw-module logging onboard module 2 environmental-history
```

Default Settings

Table 15-1 lists the default settings for OBFL parameters.

Table 15-1 Default OBFL Parameters

Parameters	Default
OBFL	All features enabled

Additional References

For additional information related to implementing OBFL, see the following sections:

- Related Documents, page 15-8
- Standards, page 15-8

Related Documents

Related Topic	Document Title
OBFL CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.x*
configuration files	*Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 4.x*
VDCs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

CHAPTER 16

Configuring NetFlow

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

This chapter includes the following sections:

Information About NetFlow

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

This section includes the following topics:

NetFlow Overview

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

Cisco NX-OS supports the Flexible NetFlow feature that enables enhanced network anomalies and security detection. Flexible NetFlow allows you to define an optimal flow record for a particular application by selecting the keys from a large collection of predefined fields. For more information on the flow records, see the "Flow Records" section on page 16-2.

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

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

- The flow has been inactive or active for too long.
- The flow cache is getting full.
- One of the counters (packets or bytes) has exceeded its maximum value.
- You have forced the flow to export.

For more information on exporters, see the "Exporters" section on page 16-2.

You define the size of the data that you want to collect for a flow using a monitor. The monitor combines the flow record and exporter with the NetFlow cache information. For more information on monitors, see the "Monitors" section on page 16-3.

Cisco NX-OS can gather NetFlow statistics in either full or sampled mode. Cisco NX-OS analyzes all packets on the interface or subinterface for full NetFlow mode. For sampled mode, you configure the sampling algorithm and rate that Cisco NX-OS analyzes packets. For more information on samplers, see the "Samplers" section on page 16-3.

Flow Records

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

- match interface input
- match interface output
- match flow direction

For more information, see the "Creating a Flow Record" section on page 16-6.

Exporters

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

- Export destination IP address
- Source interface
- UDP port number (where the collector is listening for NetFlow packets)
- Export format

Note NetFlow export packets use the IP address that is assigned to the source interface. If the source interface does not have an IP address assigned to it, the exporter will be inactive.

Cisco NX-OS exports data to the collector whenever a timeout occurs or when the flow is terminated (TCP Fin or Rst received, for example). You can configure the following timers to force a flow export:

- Active timeout—Cisco NX-OS does not remove the cache entries from the cache.
- Inactive timeout—Cisco NX-OS removes the cache entries from the cache.

Export Formats

Cisco NX-OS supports the Version 5 and Version 9 export formats. We recommend that you use the Version 9 export format for the following reasons:

- Variable field specification format
- Support for IPv6, Layer 2, and MPLS fields
- More efficient network utilization

If you configure the Version 5 export format, you have these limitations:

- Fixed field specifications
- No support for IPv6, Layer 2, or MPLS fields
- The Netflow.InputInterface and Netflow.OutputInterface represent a 16-bit I/O descriptor (IOD) of the interface.

  Note The IOD information of the interface can be retrieved using the **show system internal im info global** command.

For information about the Version 9 export format, see RFC 3954.

Note Cisco NX-OS supports UDP as the transport protocol for exports to up to two collectors.

Monitors

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

Samplers

If you are using sampled mode, you use the sampler to specify the rate at which packets are sampled. On high bandwidth interfaces, applying NetFlow processing to every single packet can result in high CPU utilization. Sampler configuration is for high-speed interfaces. You can configure samples for M out of N. For example, 100 out of every 10,000 packets are sampled.

High Availability

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

Virtualization Support

A virtual device context (VDC) is a logical representation of a set of system resources. Within each VDC, you can configure NetFlow. By default, Cisco NX-OS places you in the default VDC and any flows that you define in this mode are only available for interfaces in the default VDC.

For information about configuring VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Licensing Requirements for NetFlow

Product	License Requirement
NX-OS	NetFlow requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the NX-OS licensing scheme. For more information, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for NetFlow

NetFlow has the following prerequisite:

- You must understand the resources required on your device because NetFlow consumes additional memory and CPU resources.

If you configure VDCs, install the Advanced Services license and enter the desired VDC. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*.

Guidelines and Limitations

NetFlow has the following configuration guidelines and limitations:

- You must configure a source interface. If you do not configure a source interface, the exporter will remain in a disabled state.
- You must configure a valid record name for every flow monitor.
- A rollback will fail if you try to modify a record that is programmed in the hardware during a rollback.
- Only Layer 2 NetFlow is applied on Layer 2 interfaces, and only Layer 3 NetFlow is applied on Layer 3 interfaces.
- If you add a member to a port channel that is already configured for Layer 2 NetFlow, its NetFlow configuration is removed and the Layer 2 configuration of the port channel is added to it.

- If you change a Layer 2 interface to a Layer 3 interface, the software removes the Layer 2 NetFlow configuration from the interface.
- Use v9 export to see the full 32-bit SNMP ifIndex values at the NetFlow connector.

Configuring NetFlow

To configure NetFlow, follow these steps:

Step 1 Enable the NetFlow feature (see the "Enabling the NetFlow Feature" section on page 16-5).

Step 2 Define a flow record by specifying keys and fields to the flow (see the "Creating a Flow Record" section on page 16-6).

Step 3 Define an optional flow exporter by specifying the export format, protocol, destination, and other parameters (see the "Creating a Flow Exporter" section on page 16-9).

Step 4 Define a flow monitor based on the flow record and flow exporter (see the "Creating a Flow Monitor" section on page 16-11).

Step 5 Apply the flow monitor to a source interface, subinterface, VLAN interface (see the "Applying a Flow to an Interface" section on page 16-13), or a VLAN (see the "Configuring Bridged NetFlow on a VLAN" section on page 16-14).

This section includes the following topics:

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

Enabling the NetFlow Feature

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

Use the following command in global configuration mode to enable NetFlow:

Command	Purpose
feature netflow **Example:** `switch(config)# feature netflow`	Enables the NetFlow feature.

Use the following command in global configuration mode to disable NetFlow and remove all flows:

Command	Purpose
no feature netflow **Example:** `switch(config)# no feature netflow`	Disables the NetFlow feature. The default is disabled.

Creating a Flow Record

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **flow record** *name*
3. **description** *string*
4. **match** *type*
5. **collect** *type*
6. **show flow record** [**name**] [*record-name* | **netflow-original** | **netflow protocol-port** | **netflow** {**ipv4** | **ipv6**} {**original-input** | **original-output**}}
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

	Command	Purpose

Step 2	**flow record** *name*
	
	Example:
	`switch(config)# flow record Test`
	`switch(config-flow-record)#`
		Creates a flow record and enters flow record configuration mode.

Step 3	**description** *string* **Example:** `switch(config-flow-record)# description Ipv4Flow`	(Optional) Describes this flow record as a maximum 63-character string.
Step 4	**match** *type* **Example:** `switch(config-flow-record)# match transport destination-port`	Specifies a match key. See the "Specifying the Match Parameters" section on page 16-7 for more information on the *type* argument.
Step 5	**collect** *type* **Example:** `switch(config-flow-record)# collect counter packets`	Specifies the collection field. See the "Specifying the Collect Parameters" section on page 16-8 for more information on the *type* argument.

Step 6	**show flow record** [**name**] [*record-name* | **netflow-original** | **netflow protocol-port** | **netflow** {**ipv4** | **ipv6**} {**original-input** | **original-output**}}
	
	Example:
	`switch(config-flow-exporter)# show flow record netflow protocol-port`
		(Optional) Displays information about NetFlow flow records.

Step 7	**copy running-config startup-config** **Example:** `switch(config-flow-exporter)# copy running-config startup-config`	(Optional) Saves this configuration change.

Specifying the Match Parameters

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

Command	Purpose

match ip {**protocol** | **tos**}
 	
Example:
`switch(config-flow-record)# match ip protocol`
	Specifies the IP protocol or ToS fields as keys.

match ipv4 {**destination address** | **source address**}
 	
Example:
`switch(config-flow-record)# match ipv4 destination address`
	Specifies the IPv4 source or destination address as a key.

Command	Purpose

match ipv6 {**destination address** | **source address** | **flow-label** | **options**}
 	
Example:
`switch(config-flow-record)# match ipv6 flow-label`
	Specifies the IPv6 key.

match transport {**destination-port** | **source-port**}
 	
Example:
`switch(config-flow-record)# match transport destination-port`
	Specifies the transport source or destination port as a key.

match datalink {**mac source-address** | **mac destination-address** | **ethertype** | **vlan**}
 	
Example:
`switch(config-flow-record)# match datalink ethertype`
	Specifies the Layer 2 attribute as a key.

Specifying the Collect Parameters

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

Command	Purpose
collect counter {**bytes** | **packets**} [**long**] **Example:** `switch(config-flow-record)# collect counter packets`	Collects either packet-based or byte counters from the flow. You can optionally specify that 64-bit counters are used.

collect flow {**direction** | **sampler id**}
 	
Example:
`switch(config-flow-record)# collect flow direction`
	Collects the direction of the flow or the sampler identifier used for the flow.

collect routing {**destination** | **source**} **as** [**peer**] **Example:** `switch(config-flow-record)# collect routing destination as`	Collects the source or destination AS number of the local device or the peer.
collect routing forwarding-status **Example:** `switch(config-flow-record)# collect routing forwarding-status`	Collects the forwarding status of the packet.
collect routing next-hop address ipv4 [**bgp**] **Example:** `switch(config-flow-record)# collect routing next-hop address ipv4`	Collects the next-hop IPv4 address.

Command	Purpose
collect routing next-hop address ipv6 [**bgp**] **Example:** `switch(config-flow-record)# collect routing next-hop address ipv6`	Collects the next-hop IPv6 address.

collect timestamp sys-uptime {**first** | **last**}
Example:
`switch(config-flow-record)# collect timestamp sys-uptime last`
	Collects the system up time for the first or last packet in the flow.

collect transport tcp flags **Example:** `switch(config-flow-record)# collect transport tcp flags`	Collects the TCP transport layer flags for the packets in the flow.

Creating a Flow Exporter

You can create a flow export to define the export parameters for a flow.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **flow exporter** *name*
3. **destination** {*ipv4-address* | *ipv6-address*} [**use-vrf** *name*]
4. **source** *interface-type number*
5. **version** {**5** | **9**}
6. **show flow exporter** [*name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**flow exporter** *name* **Example:** switch(config)# flow exporter ExportTest switch(config-flow-exporter)#	Creates a flow exporter and enters flow exporter configuration mode.
Step 3	**destination** {*ipv4-address* | *ipv6-address*} [**use-vrf** *name*] **Example:** switch(config-flow-exporter)# destination 192.0.2.1	Sets the destination IPv4 or IPv6 address for this exporter. You can optionally configure the VRF to use to reach the NetFlow collector.
Step 4	**source** *interface-type number* **Example:** switch(config-flow-exporter)# source ethernet 2/1	Specifies the interface to use to reach the NetFlow collector at the configured destination.
Step 5	**version** {**5** | **9**} **Example:** switch(config-flow-exporter)# version 9 switch(config-flow-exporter-version-9)#	Specifies the NetFlow export version. Version 9 enters the export version configuration submode.
Step 6	**show flow exporter** [*name*] **Example:** switch(config-flow-exporter)# show flow exporter	(Optional) Displays information about NetFlow flow exporters.
Step 7	**copy running-config startup-config** **Example:** switch(config-flow-exporter)# copy running-config startup-config	(Optional) Saves this configuration change.

You can optionally configure the following parameters for flow exporters:

Command	Purpose
description *string* **Example:** switch(config-flow-exporter)# description ExportV9	Describes this flow exporter as a maximum 63-character string.

Command	Purpose
dscp *value* **Example:** `switch(config-flow-exporter)# dscp 0`	Specifies the differentiated services codepoint value. The range is from 0 to 63.
transport udp number **Example:** `switch(config-flow-exporter)# transport udp 200`	Specifies the UDP port to use to reach the NetFlow collector. The range is from 0 to 65535.

You can optionally configure the following parameters in flow exporter version configuration submode:

Command	Purpose
option {**exporter-stats** | **interface-table** | **sampler-table**} **timeout** *seconds* **Example:** `switch(config-flow-exporter-version-9)# option exporter-stats timeout 1200`	Sets the exporter resend timer. The range is from 1 to 86400 seconds.
template data timeout *seconds* **Example:** `switch(config-flow-exporter-version-9)# template data timeout 1200`	Sets the template data resend timer. The range is from 1 to 86400 seconds.

Creating a Flow Monitor

You can create a flow monitor and associate it with a flow record and a flow exporter.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **flow monitor** *name*
3. **description** *string*
4. **exporter** *name*
5. **record** {*name* | **netflow-original** | **netflow protocol-port** | **netflow** {**ipv4** | **ipv6**} {**original-input** | **original-output**}}
6. **show flow monitor** [*name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`flow monitor`** *`name`*
	Example:
	`switch(config)# flow monitor MonitorTest`
	`switch(config-flow-monitor)#`
		Creates a flow monitor and enters flow monitor configuration mode.

Step 3	**`description`** *`string`*
	Example:
	`switch(config-flow-monitor)# description Ipv4Monitor`
		(Optional) Describes the flow monitor with an alphanumeric string up to 63 characters.

Step 4	**`exporter`** *`name`*
	Example:
	`switch(config-flow-monitor)# exporter Exportv9`
		Associates a flow exporter with this flow monitor.

Step 5	**`record`** {*`name`* | **`netflow-original`** | **`netflow protocol-port`** | **`netflow`** {**`ipv4`** | **`ipv6`**} {**`original-input`** | **`original-output`**}}
	Example:
	`switch(config-flow-monitor)# record IPv4Flow`
		Associates a flow record with the specified flow monitor.

Step 6	**`show flow monitor`** [*`name`*] **Example:** `switch(config-flow-monitor)# show flow monitor`	(Optional) Displays information about NetFlow flow monitors.

Step 7	**`copy running-config startup-config`**
	Example:
	`switch(config-flow-monitor)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Creating a Sampler

You can create a sampler to define the NetFlow sampling rate for a flow.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **sampler** *name*
3. **description** *string*
4. **mode** *samples* **out-of** *packets*
5. **show sampler** [*name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`sampler`** *`name`*
	
	Example:
	`switch(config)# sampler SampleTest`
	`switch(config-flow-sampler)#`
		Creates a sampler and enters flow sampler configuration mode.

Step 3	**`description`** *`string`*
	
	Example:
	`switch(config-flow-sampler)# description Samples`
		(Optional) Describes the sampler with an alphanumeric string up to 63 characters.

Step 4	**`mode`** *`samples`* **`out-of`** *`packets`*
	
	Example:
	`switch(config-flow-sampler)# mode 1 out-of 100`
		Defines the number of samples to take per the number of packets received. The samples range is from 1 to 64. The packets range is from 1 to 8192 packets.

Step 5	**`show sampler`** [*`name`*] **Example:** `switch(config-flow-sampler)# show sampler`	(Optional) Displays information about NetFlow samplers.

Step 6	**`copy running-config startup-config`**
	
	Example:
	`switch(config-flow-sampler)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Applying a Flow to an Interface

You can apply a flow monitor and an optional sampler to an interface.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **interface** *interface-type number*
3. **ip flow monitor** *name* {**input** | **output**} [**sampler** *name*]
4. **ipv6 flow monitor** *name* {**input** | **output**} [**sampler** *name*]
5. **show flow interface** [*interface-type number*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`interface`** *`interface-type number`*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode. The interface type can be Ethernet (including subinterfaces), port channel, VLAN, SVI, or tunnel.

Step 3	**`ip flow monitor`** *`name`* {**`input`** | **`output`**} [**`sampler`** *`name`*] **Example:** `switch(config-if)# ip flow monitor MonitorTest input`	Associates an IPv4 flow monitor and an optional sampler to the interface for input or output packets.
Step 4	**`ipv6 flow monitor`** *`name`* {**`input`** | **`output`**} [**`sampler`** *`name`*] **Example:** `switch(config-if)# ipv6 flow monitor MonitorTest input`	Associates an IPv6 flow monitor and an optional sampler to the interface for input or output packets.
Step 5	**`show flow interface`** [*`interface-type number`*] **Example:** `switch(config-if# show flow interface`	(Optional) Displays information about NetFlow on an interface.

Step 6	**`copy running-config startup-config`**
	Example:
	`switch(config-if)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Configuring Bridged NetFlow on a VLAN

You can apply a flow monitor and an optional sampler to a VLAN.

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **vlan** *vlan-id*
3. **ip flow monitor** *name* {**input** | **output**} [**sampler** *name*]
4. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`config t`**
	
	Example:
	`switch# config t`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch(config)#`
		Places you in global configuration mode.

Step 2	**`vlan`** *`vlan-id`*
	
	Example:
	`switch(config)# vlan 30`
	`switch(config-vlan)#`
		Enters VLAN configuration mode. The *vlan-id* range is from 1 to 3967 or from 4048 to 4093.

Step 3	**`ip flow monitor`** *`name`* `{`**`input`** `|` **`output`**`}` `[`**`sampler`** *`name`*`]`
	
	Example:
	`switch(config-vlan)# ip flow monitor MonitorTest input`
		Associates a flow monitor and an optional sampler to the VLAN for input or output packets.

Step 4	**`copy running-config startup-config`**
	
	Example:
	`switch(config-vlan)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Configuring Layer 2 NetFlow

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

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

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

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

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

BEFORE YOU BEGIN

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

SUMMARY STEPS

1. **config t**
2. **flow record** *name*
3. **match datalink** {**mac source-address** | **mac destination-address** | **ethertype** | **vlan**}
4. **interface** {**ethernet** *slot*/*port*} | {**port-channel** *number*}
5. **switchport**
6. **mac packet-classify**
7. **layer2-switched flow monitor** *flow-name* **input** [**sampler** *sampler-name*]
8. **show flow record netflow layer2-switched input**
9. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**config t** **Example:** switch# config t Enter configuration commands, one per line. End with CNTL/Z. switch(config)#	Places you in global configuration mode.
Step 2	**flow record** *name* **Example:** switch(config)# flow record L2_record	Enters flow record configuration mode. For more information about configuring flow records, see the "Creating a Flow Record" section on page 16-6.
Step 3	**match datalink** {**mac source-address** | **mac destination-address** | **ethertype** | **vlan**} **Example:** switch(config-flow-record)# match datalink ethertype	Specifies the Layer 2 attribute as a key.
Step 4	**interface** {**ethernet** *slot*/*port*} | {**port-channel** *number*} **Example 1:** switch(config)# interface ethernet 2/1 switch(config-if)# **Example 2:** switch(config)# interface port-channel 8 switch(config-if)#	Enters interface configuration mode. The interface type can be a physical Ethernet port or a port channel.

	Command	Purpose
Step 5	**switchport** **Example:** switch(config-if)# switchport	Changes the interface to a Layer 2 physical interface. For information about configuring switch ports, see the *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 4.x*.
Step 6	**mac packet-classify** **Example:** switch(config-if)# mac packet-classify	Forces MAC classification of packets. For more information about using the **mac packet-classify** command, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.x*.
Step 7	**layer2-switched flow monitor** *flow-name* **input** [**sampler** *sampler-name*] **Example:** switch(config-vlan)# layer2-switched flow monitor L2_monitor input sampler L2_sampler	Associates a flow monitor and an optional sampler to the switch port input packets. For information about flow monitors, see the "Creating a Flow Monitor" section on page 16-11. For information about samplers, see the "Creating a Sampler" section on page 16-12.
Step 8	**show flow record netflow layer2-switched input** **Example:** switch(config-if# show flow record netflow layer2-switched input	(Optional) Displays information about the Layer 2 Netflow default record.
Step 9	**copy running-config startup-config** **Example:** switch(config-vlan)# copy running-config startup-config	(Optional) Saves this configuration change.

Configuring NetFlow Timeouts

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

Use the following commands in global configuration mode to configure NetFlow timeout parameters:

Command	Purpose
flow timeout active *seconds* **Example:** switch(config)# flow timeout active 90	Sets the active timeout value in seconds. The range is from 60 to 4092. The default is 1800.
flow timeout aggressive threshold *percent* **Example:** switch(config)# flow timeout aggressive threshold 90	Enables using a percentage that you want the NetFlow table to be before aggressive aging starts. The range is from 50 to 99. The default is disabled.
flow timeout fast *seconds* **threshold** *packets* **Example:** switch(config)# flow timeout fast 40 threshold 1200	Enables using a fast timeout value and the number of packets in a flow before aging begins. The fast timeout range in seconds is from 32 to 512. The packet range is from 1 to 4000. The default is disabled.

Command	Purpose
flow timeout inactive *seconds* **Example:** `switch(config)# flow timeout inactive 900`	Sets the inactive timeout value in seconds. The range is from 15 to 4092. The default is 15.
flow timeout session **Example:** `switch(config)# flow timeout session`	Enables TCP session aging. The default is disabled.

Verifying NetFlow Configuration

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

Command	Purpose
show flow exporter [*name*]	Displays information about NetFlow flow exporters and statistics.
show flow interface [*interface-type number*]	Displays information about NetFlow interfaces.
show flow monitor [*name*] [**cache** [**detailed**]]	Displays information about NetFlow flow monitors and statistics.
show flow record [*name*]	Displays information about NetFlow flow records.
show flow record netflow layer2-switched input	Displays information about the Layer 2 NetFlow configuration.
show flow timeout	Displays information about NetFlow timeouts.
show hardware flow aging [**vdc** *vdc_id*] [**detail**] [**module** *module*]	Displays information about NetFlow aging flows in the hardware.
show hardware flow entry address *table-address* **type** {*ip* | *ipv6*} [**module** *module*]	Displays information about NetFlow table entries in the hardware.
show hardware flow ip [interface *type number* | **monitor** *monitor_name* | **profile** *profile-id* | **vdc** *vdc_id* | **vlan** *vlan_id*] [**detail**] [**module** *module*]	Displays information about NetFlow IPv4 flows in the hardware.
show hardware flow sampler [**all** | **count** | **index** *number* | **name** *sampler-name* | **vdc** *vdc_id*] [**detail**] [**module** *module*]	Displays information about the NetFlow sampler in the hardware.
show hardware flow utilization [**module** *module*]	Displays information about NetFlow table utilization in the hardware.
show sampler [*name*]	Displays information about NetFlow samplers.

Monitoring NetFlow

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

Use the **clear flow exporter** command to clear NetFlow exporter statistics. Use the **clear flow monitor** command to clear the monitor cache and statistics.

NetFlow Example Configuration

This example shows how to create a flow and apply it to an interface:

```
feature netflow
flow exporter ee
 version 9
flow record rr
 match ipv4 source address
 match ipv4 destination address
 collect counter bytes
 collect counter packets
flow monitor foo
 record rr
 exporter ee
interface Ethernet2/45
 ip flow monitor foo output
 ip address 10.20.1.1/24
 no shutdown
```

Default Settings

Table 16-1 lists the default settings for NetFlow parameters.

Table 16-1 Default NetFlow Parameters

Parameters	Default
Egress and Ingress cache size	512K
Flow active timeout	1800 seconds
Flow timeout aggressive threshold	disabled
Flow timeout fast threshold	disabled
Flow timeout inactive	15 seconds
Flow timeout session aging	disabled

Additional References

For additional information related to implementing NetFlow, see the following sections:

- Related Documents, page 16-20
- Standards, page 16-20

Related Documents

Related Topic	Document Title
NetFlow CLI commands	*Cisco Nexus 7000 Series NX-OS System Management Command Reference*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 4.x*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for NetFlow

Table 16-2 lists the release history for this feature.

Table 16-2 Feature History for Rollback

Feature Name	Releases	Feature Information
Layer 2 NetFlow	4.2(1)	You can define Layer 2 keys in flexible NetFlow records that you can use to capture flows in Layer 2 interfaces. See the "Guidelines and Limitations" section on page 16-4. See the "Configuring Layer 2 NetFlow" section on page 16-15.
Rollback during NetFlow	4.1(3)	Rollback fails for NetFlow if during rollback, you try to modify a record that is programmed in the hardware. See the "Guidelines and Limitations" section on page 16-4.

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.

RFCs

RFCs	Title
RFC 2819	*Remote Network Monitoring Management Information Base*
RFC 3164	*The BSD syslog Protocol*
RFCs 3411 to 3418	*An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*
RFC 3954	*Cisco Systems NetFlow Services Export Version 9*

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, page B-1
- EEM Events, page B-3
- EEM Policy Configuration Examples, page B-3

EEM System Policies

Table B-1 lists the Embedded Event Manger (EEM) system policies.

Table B-1 EEM System Policies

Event	Description
__PortLoopback	Do CallHome, log error in Syslog/OBFL/Exception Log and disable further HM testing on affected ports after 20 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 20 consecutive failures of GOLD "RewriteEngine" test
__asic_register_check	Do CallHome, log error and disable further HM testing for that ASIC device/instance after 20 consecutive failures of GOLD "ASICRegisterCheck" test
__compact_flash	Do CallHome, log error and disable further HM testing after 20 consecutive failures of GOLD "CompactFlash" test
__crypto_device	Do CallHome and log error when GOLD "CryptoDevice" test fails
__eobc_port_loopback	Do CallHome and log error when GOLD "EOBCPortLoopback" test fails
__ethpm_debug_1	Action: none

Table B-1 EEM System Policies (continued)

Event	Description
__ethpm_link_flap	Too many link flaps in a short interval. Action: Error Disable the port
__external_compact_flash	Do CallHome, log error and disable further HM testing after 20 consecutive failures of GOLD "ExternalCompactFlash" test
__lcm_module_failure	Power-cycle 2 times 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	Shutdown if both fans (f1 & f2) are together absent for two minutes
__pfm_fanabsent_any_singlefan	Shutdown half-chassis if fan absent for three minutes
__pfm_fanbad_all_systemfan	Shutdown if both fans (f1 & f2) are together bad for two minutes
__pfm_fanbad_any_singlefan	Syslog when fan goes bad
__pfm_power_over_budget	Syslog warning for insufficient power over budget
__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 themodule 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
__status_bus	Do CallHome, log error and disable further HM testing after 5 consecutive failures of GOLD "StatusBus" test
__standby_fabric_loopback	Do CallHome, log error in Syslog/OBFL/ ExceptionLog and disable further HM testing for that module after 10 consecutive failures
__system_mgmt_bus	Do Call Home, log error and disable further HM testing for that FAN/PS after 20 consecutive failures
__usb	Do Call Home and log error

EEM Events

Table B-2 describes the EEM events you can use on the device.

Table B-2 EEM Events

EEM Event	Description
cli	CLI command is entered that matches a pattern with a wildcard.
counter	EEM counter reaches a specified value or range.
fanabsent	System fan is absent.
fanbad	System fan generates a fault.
gold	GOLD test failure condition is hit.
memory	Available system memory exceeds a threshold.
module-failure	Module failure is generated.
oir	Online insertion or removal occurs.
policy-default	Default parameters and thresholds are used for the events in the system policy you override.
poweroverbudget	Platform software detects a power budget condition.
snmp	SNMP object ID (OID) state changes.
storm-control	Platform software detects an Ethernet packet storm condition.
sysmgr	System manager generates an event.
temperature	Temperature level in the system exceesd a threshold.
track	Tracked object changes state.

EEM Policy Configuration Examples

This section includes the following topics:

- Configuration Examples for CLI Events, page B-4
- Configuration Examples to Override (Disable) Major Thresholds, page B-5
- Configuration Examples to Override (Disable) Shutdown for Fan Removal, page B-8
- Configuration Examples to Create a Supplemental Policy, page B-10
- Configuration Examples for the Power Over-Budget Policy, page B-11
- Configuration Examples to Select Modules to Shut Down, page B-12
- Configuration Examples for the Online Insertion Removal Event, page B-13
- Configuration Example to Generate a User syslog, page B-13
- Configuration Examples for SNMP Notification, page B-13
- Configuration Example for Port Tracking, page B-14

Configuration Examples for CLI Events

This section includes the following examples of CLI event configuration:

- Monitoring Interface Shutdown, page B-4
- Monitoring Module Powerdown, page B-4
- Adding a Trigger to Initiate a Rollback, page B-4

Monitoring Interface Shutdown

This example shows how to monitor an interface shutdown:

```
switch# config t
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# config t
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# config t
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

This section includes the following topics:

- Preventing a Shutdown When Reaching a Major Threshold, page B-5
- Disabling One Bad Sensor, page B-5
- Disabling Multiple Bad Sensors, page B-6
- Overriding (Disabling) an Entire Module, page B-6
- Overriding (Disabling) Multiple Modules and Sensors, page B-6
- Enabling One Sensor While Disabling All Remaining Sensors of All Modules, page B-7
- Enabling Multiple Sensors While Disabling All Remaining Sensors of All Modules, page B-7
- Enabling All Sensors of One Module While Disabling All Sensors of the Remaining Modules, page B-7
- Enabling a Combination of Sensors on Modules While Disabling All Sensors of the Remaining Modules, page B-8

Preventing a Shutdown When Reaching a Major Threshold

This example shows how to prevent a shutdown caused by reaching a major threshold:

```
switch# config t
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

This example shows how to revert to the default configuration:

```
switch# config t
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# config t
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config0aooket)# end
```

This example shows how to revert to the default configuration:

```
switch# config t
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# config t
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# config t
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# config t
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
switch(config-epplet)# end
```

This example shows how to revert to the default configuration:

```
switch# config t
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# config t
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# config t
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# config t
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# config t
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# config t
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# config t
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# config t
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# config t
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# config t
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# config t
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# config t
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# config t
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# config t
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# config t
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# config t
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 modules 3 and 4:

```
switch# config t
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# config t
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# config t
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# config t
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# config t
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 Removal

This section includes the following topics:

- Overriding (Disabling) a Shutdown for Removal of One or More Fans, page B-8
- Overriding (Disabling) a Shutdown for Removal of a Specified Fan, page B-9
- Overriding (Disabling) a Shutdown for Removal of Multiple Specified Fans, page B-9
- Overriding (Disabling) a Shutdown for Removal of All Fans Except One, page B-9
- Overriding (Disabling) a Shutdown for Removal of Fans Except for a Specified Set of Fans, page B-10
- Overriding (Disabling) a Shutdown for Removal of All Fans Except One from a Set of Fans, page B-10

Overriding (Disabling) a Shutdown for Removal of One or More Fans

This example shows how to disable a shutdown so that you can remove one or more (or all) fans:

```
switch# config t
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# config t
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

This example shows how to disable a shutdown so that you can remove a specified fan (fan 3):

```
switch# config t
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# config t
switch(config) no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config)# end
```

Overriding (Disabling) a Shutdown for Removal of Multiple Specified Fans

This example shows how to disable a shutdown so that you can remove multiple specified fans (fans 2, 3, and 4):

```
switch# config t
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# config t
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# config t
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# config t
switch(config)# no event manager applet myappletname override
__pfm_fanabsent_any_singlefan
switch(config)# end
```

Overriding (Disabling) a Shutdown for Removal of All Fans Except One

This example shows how to disable a shutdown so that you can remove all fans except one (fan 2):

```
switch# config t
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch# config t
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 Fans Except for a Specified Set of Fans

This example shows how to disable a shutdown so that you can remove fans except for a specified set of fans (fans 2, 3, and 4):

```
switch# config t
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 Fans Except One from a Set of Fans

This example shows how to disable a shutdown so that you can remove all fans except one from a set of fans (fans 2, 3, or 4):

```
switch# config t
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch# config t
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# config t
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# config t
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

This section includes the following topics:

- Creating a Supplemental Policy for the Fan Absent Event, page B-10
- Creating a Supplemental Policy for the Temperature Threshold Event, page B-11

Creating a Supplemental Policy for the Fan Absent Event

This example shows how to create a supplemental policy using the **event fanabsent** command:

[**no**] **event fanabsent** [**fan** *fan-number*] **time** *time-interval*

In addition to the default policy, this example shows how to execute the policy myappletname and action 3 if fan 1 is absent for 60 seconds:

```
switch# config t
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# config t
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.

This section includes the following topics:

- Shutting Down Modules, page B-11
- Shutting Down a Specified List of Modules, page B-12

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# config t
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# config t
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

This section includes the following topics.:

- Using the Policy Default to Select Nonoverridden Modules to Shut Down, page B-12
- Using Parameter Substitution to Select Nonoverridden Modules to Shut Down, page B-12

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# config t
switch(config)# event manager applet my5a1 override __pfm_tempev_major
switch(config-applet)# end
switch# config t
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# config t
switch(config)# event manager applet my5b1 override __pfm_tempev_major
switch(config-applet)# end
switch# config t
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# config t
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# config t
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# config t
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 will generate a syslog as follows:

p1b-57(config)# 2008 Feb 20 00:08:27 p1b-57 %$ VDC-1 %$ %EEM_ACTION-2-CRIT: “Module is removed”

2008 Feb 20 00:08:27 p1b-57 %$ VDC-1 %$ %PLATFORM-2-MOD_REMOVE: Module 2 removed (Serial number JAB120101PW)

Configuration Examples for SNMP Notification

This section includes the following topics:

- Polling an SNMP OID to Generate an EEM Event, page B-14
- Sending an SNMP Notification in Response to an Event in the Event Policy, page B-14

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# config t
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 port tracking of Ethernet interface 3/23 by Ethernet interface 1/2, follow these steps:

Step 1 Create an object to track the status of Ethernet interface 3/23.

```
switch# config t
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.

```
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.

```
switch# config t
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
```

INDEX

Send document comments to nexus7k-docfeedback@cisco.com.