Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide

First Published: 2016-12-21

Last Modified: 2019-10-31

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

THE 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.

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

CONTENTS

Preface

This preface describes the audience, organization and conventions of the *Cisco Nexus 7706 Hardware Installation Guide*. It also provides information on how to obtain related documentation.

Preface

This preface describes the audience, organization, and conventions of the Book Title. It also provides information on how to obtain related documentation.

This chapter includes the following topics:

Audience

This publication is for experienced network administrators who configure and maintain Cisco NX-OS on Cisco Nexus 7000 Series Platform switches.

Document Conventions

Note

- As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.
- The Guidelines and Limitations section contains general guidelines and limitations that are applicable to all the features, and the feature-specific guidelines and limitations that are applicable only to the corresponding feature.

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.

Convention	Description
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information 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

Documentation for Cisco Nexus 7000 Series Switches is available at:

- Configuration Guides

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-installation-and-configuration-guides-list.html

- Command Reference Guides

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-command-reference-list.html

- Release Notes

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-release-notes-list.html

- Install and Upgrade Guides

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-installation-guides-list.html

- Licensing Guide

  http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-licensing-information-listing.html

Documentation for Cisco Nexus 7000 Series Switches and Cisco Nexus 2000 Series Fabric Extenders is available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-2000-series-fabric-extenders/products-installation-and-configuration-guides-list.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus7k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 7000 Series NX-OS Fundamentals Guide, Release 6.x*. The latest version of this document is available at the following Cisco website:

http://www.cisco.com/en/US/products/ps9402/products_installation_and_configuration_guides_list.html

New and Changed Information

To check for additional information about Cisco NX-OS Release 8.x, see the *Cisco Nexus 7000 Series NX-OS Release Notes, Release 8.x* available at the following Cisco website:

This table summarizes the new and changed features for the *Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide*.

The table below summarizes the new and changed features for this document and shows the releases in which each feature is supported. Your software release might not support all the features in this document. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release.

Table 1: New and Changed Features

Feature Name	Description	Changed in Release
Distributed Packet Tracer	Distributed Packet Tracer (DPT) is a utility integrated within Cisco Nexus 7000/7700 platforms that can be used to trace the path of the packet through the switch.	8.2(1)

Feature Name	Description	Changed in Release
Network Plug and Play	Network plug and play (PnP) is a software application that runs on a Cisco Nexus 7000 switch. The PnP feature provides a simple, secure, unified, and integrated offering to ease new branch or campus roll-outs, and for provisioning updates to an existing network. This feature provides a unified approach to provision networks that comprise different devices with a near zero-touch deployment experience.	8.2(1)
Configure Replace	The Configure Replace (CR) feature enables the Nexus switch to replace the running-configuration with the user provided configuration without reloading the device.	8.2(1)
Consistency Checker	M3 module support for interface-properties, link state and L3 interface, and consistency checker for all modules are introduced.	8.2(1)
Consistency Checker	This feature was introduced. Consistency Checker for Fabricpath FTAG-state, Fabricpath gpc-membership, Interface Properties, l2mcast, l3-interface, link-state, proxy rpc-membership, and stp-state are introduced.	8.0(1)
Fault Management System	The Fault Management System is used to enhance the Cisco NX-OS serviceability by providing an efficient means to capture data relevant and adequate to debug the issues being reported at the earliest possible time, without any manual intervention. If all the nodes are down, the packets get routed automatically.	8.0(1)
63 character hostname and switch name	Supports 63 characters for hostname and switch name	7.3(0)D1(1)
EXEC banner	Supports the EXEC banner feature	7.3(0)D1(1)
PowerOn Auto Provisioning (POAP) support	Automates the process of upgrading software images and installing configuration files on Cisco Nexus switches	6.1(2)

CHAPTER 2

Overview

This chapter provides an overview of the Cisco NX-OS software.

Licensing Requirements

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

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Software Compatibility

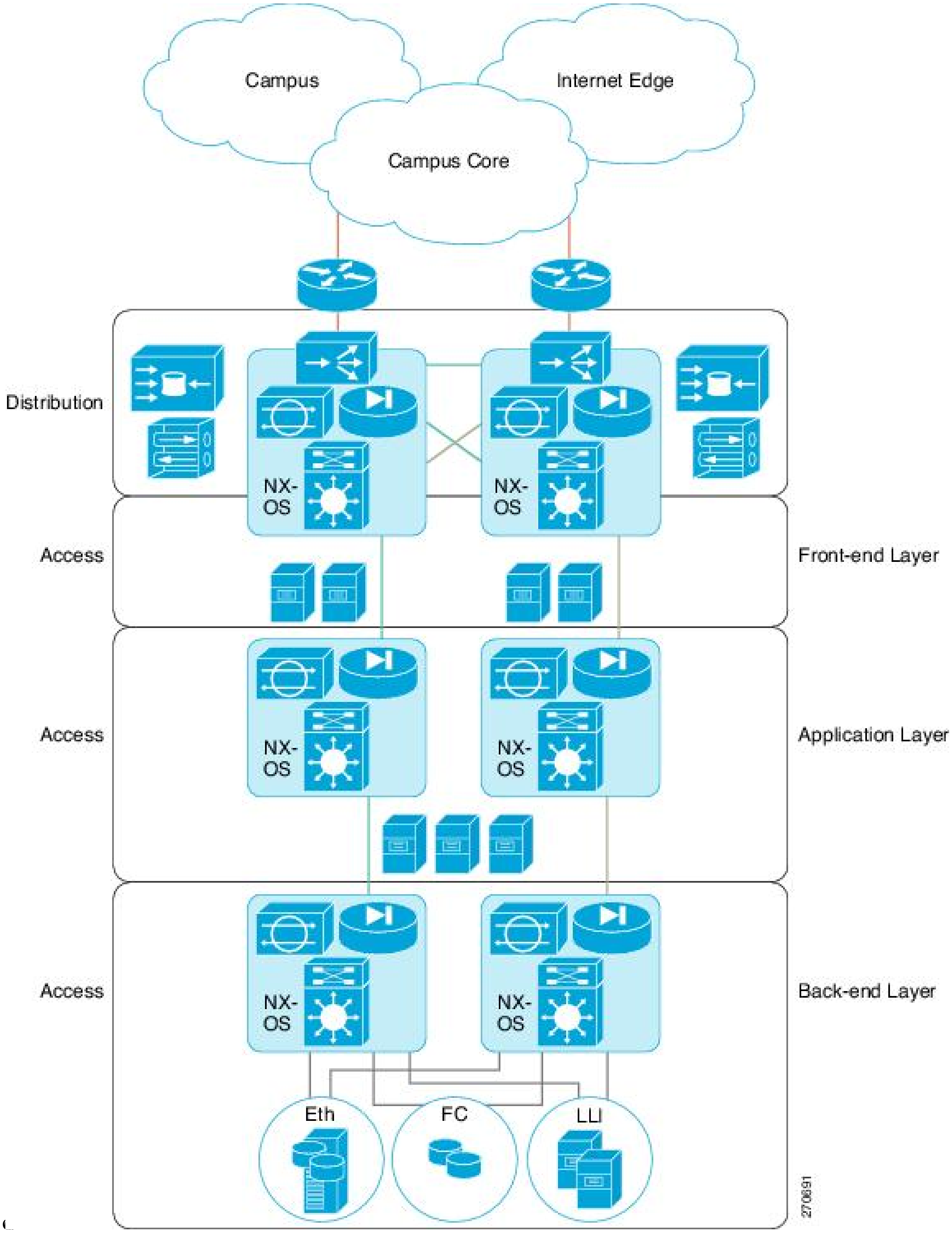
The Cisco NX-OS software interoperates with Cisco products that run any variant of the Cisco IOS software. The Cisco NX-OS software also interoperates with any networking operating system that conforms to the IEEE and RFC compliance standards.

Common Software Throughout the Data Center

The Cisco NX-OS software provides a unified operating system that is designed to run all areas of the data center network including the LAN and Layer 4 through Layer 7 network services.

Figure 1: Cisco NX-OS in a Data Center

This figure shows an overview of the Cisco NX-OS software in the data



Modular Software Design

The Cisco NX-OS software supports distributed multithreaded processing on symmetric multiprocessors (SMPs), multi-core CPUs, and distributed data module processors. The Cisco NX-OS software offloads computationally intensive tasks, such as hardware table programming, to dedicated processors distributed across the data modules. The modular processes are created on demand, each in a separate protected memory space. Processes are started and system resources are allocated only when you enable a feature. A real-time preemptive scheduler helps to ensure the timely processing of critical functions.

Virtual Device Contexts

The Cisco NX-OS software can segment system and hardware resources into virtual contexts that emulate virtual devices. Each virtual device context (VDC) has its own software processes, dedicated hardware resources (interfaces), and an independent management environment. With VDCs, you can consolidate separate networks onto a common infrastructure, which maintains the administrative boundary separation and fault isolation characteristics of physically separate networks, and provides many of the operational cost benefits of a single infrastructure. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Serviceability

The Cisco NX-OS software has serviceability functions that allow the device to respond to network trends and events. These features help you with network planning and improving response times.

Switched Port Analyzer

The Switched Port Analyzer (SPAN) feature allows you to analyze all traffic between ports (called the SPAN source ports) by nonintrusively directing the SPAN session traffic to a SPAN destination port that has an external analyzer attached to it. For more information about SPAN, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethanalyzer is a command-line version of Wireshark for capturing and decoding packets. You can use Ethanalyzer to troubleshoot your network and analyze the control-plane traffic. For more information about Ethanalyzer, see the *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide*.

Call Home

The Call Home feature continuously monitors hardware and software components to provide e-mail-based notification of critical system events. A versatile range of message formats is available for optimal compatibility with pager services, standard e-mail, and XML-based automated parsing applications. It offers alert grouping capabilities and customizable destination profiles.You can use this feature, for example, to directly page a network support engineer, send an e-mail message to a network operations center (NOC), and employ Cisco AutoNotify services to directly generate a case with the Cisco Technical Assistance Center (TAC). For more information about Call Home, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Online Diagnostics

Cisco generic online diagnostics (GOLD) verify that hardware and internal data paths are operating as designed. Boot-time diagnostics, continuous monitoring, and on-demand and scheduled tests are part of the Cisco GOLD feature set. GOLD allows rapid fault isolation and continuous system monitoring. For information about configuring GOLD, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide.*

Embedded Event Manager

Cisco Embedded Event Manager (EEM) is a device and system management feature that helps you to customize behavior based on network events as they happen. For information about configuring EEM, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

NetFlow

The Cisco NX-OS NetFlow implementation supports version 5 and version 9 exports. It also supports the Flexible NetFlow configuration model and hardware-based Sampled NetFlow for enhanced scalability. For more information about NetFlow, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Consistency Checker

Consistency Checker — Cisco NX-OS Release 8.2(1)

This section describes how to use the Consistency Checker CLIs to collect information on various table states within the software and the hardware for Cisco NX-OS Release 8.2(1).

Consistency checker compares the software state of the supervisor, with the hardware state of supported I/O modules. If there is any inconsistency, it flags the issue immediately. This helps to reduce increased troubleshooting time at a later period. Consistency checker supplements basic troubleshooting, and helps to identify scenarios where inconsistent state between software and hardware tables are causing issues in the network, thereby reducing the mean time to resolve the issue.

Consistency Checker is a serviceability tool that performs the following functions:

- Checks for consistency between software and hardware tables.
- Alerts administrators upon finding any inconsistencies.
- Helps to speed up fault isolation.

The Consistency Checker feature verifies the consistency between the software and the hardware for the following parameters in Cisco NX-OS Release 8.2(1). Except for Persistent Storage Service (PSS) consistency checker, all other features are supported since Cisco NX-OS Release 8.0(1) and are enhanced in Cisco NX-OS Release 8.2(1). Consistency checker is supported on M3 and F3 modules. Users can execute the **show consistency-checker all** command to perform consistency check for all components/features.

The following consistency checker components are supported in Cisco NX-OS Release 8.2(1):

- FabricPath

- Interface-properties
- Layer 2 Unicast
- Layer 2 Multicast
- L3-Interface Tables
- Link-state
- Proxy Forwarding
- Spanning-Tree
- Persistent Storage Service (PSS)

FabricPath

The FabricPath Consistency Checker verifies the programming consistency for the following FabricPath parameters:

- FTAG-state
- GPC-membership (Gateway Port-Channel, which is used internally for FabricPath forwarding, and this does not refer to the user-configured port-channels).

Interface-properties

The Interface-properties Consistency Checker verifies the programming consistency between software and hardware for EthPM tables (Ethernet Port Manager) including the following parameters:

- Link state
- Interface MTU
- Flow control
- FEX fabric port
- Native VLAN

Layer 2 Unicast

The Layer 2 Unicast Consistency Checker verifies the programming consistency between software and hardware tables for classical Ethernet (CE) Layer 2 unicast mac address entries.

Layer 2 Multicast

The Layer 2 Multicast Consistency Checker verifies the programming consistency between software and hardware tables for Layer 2 IGMP snooping entries in classical Ethernet (CE) topologies.

L3-Interface Tables

The L3-Interface Consistency Checker verifies the programming consistency between software and hardware for Layer 3-interface ingress and egress forwarding tables.

L3-interace consistency checker is supported only on the M3 and F3 VDCs in Cisco NX-OS Release 8.2(1). It is not supported on the VDC combination that contains a module other than M3 or F3.

Link-state

The Link-state Consistency Checker verifies the programming consistency between software and hardware for the link-state status of the interfaces.

Spanning-Tree

The Spanning-Tree Consistency Checker verifies the programming consistency between software and hardware tables for the Spanning-Tree state.

Persistent Storage Service (PSS)

The PSS Consistency Checker verifies the consistency between run-time data and data stored in PSS for the following parameters:

- Spanning-Tree
- Various ingress and egress forwarding parameters for interfaces (ELTM)
- Interface state (ETHPM)
- VLAN information (Vlan-manager)
- vPC state (vPC manager)

PSS Consistency Checker checks the system state before and after system triggers (switch over, reload, and ISSU). Invoke PSS consistency checker in steady state to avoid false alarms.

Guidelines and Limitations

- Consistency checkers are supported only on M3 and F3 Modules. Only F3 modules are supported in Cisco NX-OS Release 8.0(x), and Cisco NX-OS Release 8.1(x) releases.
- If there is a configuration change or a table state change in the environment while a consistency checker is running, it is possible to trigger false positives. In cases where false positives may be a concern, it is recommended to run multiple iterations of that consistency checker.
- L3-interface consistency checker supports only L3 standalone, L3 port channel IPv4 and IPv6 interfaces, and L3 FEX HIF interfaces. Logical interfaces such as OTV, NVE, and tunnel are not supported.
- Layer 2 multicast consistency checker supports only CE (classical Ethernet) IGMP Snooping entries. VxLAN, OTV, and Fabricpath entries for example, are not supported. Layer 2 multicast consistency checker cannot be used when unsupported features such as Fabricpath/ EVPN) is enabled on a VDC.

Using the Consistency Checker CLIs

To verify the consistency between the hardware and software for the Consistency Checker parameter for Cisco NX-OS Release 8.2(1) uses the following CLIs:

Command	Purpose
show consistency-checker link-state	Verifies the programming consistency between software and hardware for the link-state status of the interfaces.
show consistency-checker interface-properties module *[module number]*	Verifies the interface properties for all modules. Use the *[module]* keyword to verify the properties for a specific module.
show consistency-checker stp-state	Verifies the programming consistency between software and hardware tables for the Spanning-Tree state.
show consistency-checker l2mcast { *vlan ID* } { *group address* | *source address* } [all] [detail]	Verifies the layer-2 multicast consistency for L2 IGMP Snooping entries between supervisor and I/O modules

show consistency-checker l3-interface { *if index* | **bdi** | **ethernet** | **port-channel** }	Verifies the programming consistency between software and hardware for L3-interface ingress and egress forwarding tables
show consistency-checker fabricpath {ftag-state | gpc-membership}	Verifies the ftag CBL state in the software and the hardware and the FabricPath gateway port-channel membership.
show consistency-checker proxy rpc membership	Verifies the proxy router port-channel membership.
show consistency-checker l2unicast *module number*	Verifies consistency for L2 mac address table between supervisor software and I/O module hardware
show consistency-checker pss	Verifies the consistency between run-time data and data stored in PSS for STP, ELTM, ETHPM, VLAN manager, and vPC manager.
show consistency-checker all	Performs all available consistency checkers.

Consistency Checker — Cisco NX-OS Release 8.0(1)

The following sections are applicable for Cisco NX-OS Release 8.0(1).

Consistency Checker is a serviceability tool that performs the following functions:

- Checks for system consistency
- Helps perform root cause analysis and fault isolation
- Checks for consistency between software and hardware tables
- Performs on-demand trigger through CLI or NX-API

Consistency Checker consists of the following components:

- **Ethernet Port Manager (EthPM)**—Provides software values for the following parameters:
    - Link state—Provides software support on Ethernet interfaces, Fabric Extender (FEX) interfaces, and breakout interfaces.
    - Flow control—Provides software support on Ethernet interfaces, FEX interfaces, breakout interfaces, and port-channel interfaces.
    - FEX fabric port or any other port—Provides software support on FEX fabric port or any other port.
    - Native VLAN—Provides software support on L2 Ethernet interfaces, L2 FEX interfaces, L2 breakout interfaces, and L2 port-channel interfaces.
- **Spanning Tree Protocol (STP)**—Checks logical port-state consistency, either port or VLAN. Consistency is checked against STP and PIXM components.

Note Currently, consistency is checked only against the STP internal database based on the software port state and from the response provided by the PIXM on any port-state request.

- **PIXM**—Establishes relationship between the following parameters:

- Port-channel membership between PIXM and port channel
- Gateway port channel (GPC) membership between Private Internet Exchange Manager (PIXM) and Multi Channel Manager (MCM)
- RPC membership between PIXM and MCM
- VLAN CBL membership between STP, PIXM, and HW
- FTAG CBL membership between PIXM and HW

- **L2MCAST**—Verifies Layer 2 multicast (L2MCAST) route consistency across Internet Group Management Protocol (IGMP), Multicast Layer 2 RIB (M2RIB), Multicast FIB (MFIB) Distribution (MFDM), PIXM, and L2MCAST.

Note Currently, L2MCAST supports only Classical Ethernet (CE) mode and not FabricPath.

- **L3 interface properties**—Checks consistency between the contents of various forwarding hardware tables (LDB, ILM, ELM, PVV, and so on) used in L3 interfaces and their expected contents that are stored in ELTM or IFTMC. Consistency is checked on L3 interfaces, L3 port channels, L3 FEX ports, L3 HIF port channels, and L3 interface VLANs.

Output Examples for Consistency Checker Components

Output Examples for Consistency Checker Components – Cisco NX-OS Release 8.2(1)

Example: Show Consistency Checker All Output

```
switch# show consistency-checker all

--------------------------------------------------------------
Consistency checker started at 2017 Sep 29 20:54:09 .
Please run 'show consistency-checker all status' to see the status.
--------------------------------------------------------------
switch# show consistency-checker all status
--------------------------------------------------------------
Consistency checker was started at 2017 Sep 29 20:54:09 .
Consistency checker in progress !
--------------------------------------------------------------
switch# show consistency-checker all output
Consistency-checker result:
(VDC: 1 ,TIME: 2017 Sep 29 20:54:09)
-------------------------------------------------------------
Consistency Checker Result for Ftag CBL: SUCCESS
-------------------------------------------------------------
-------------------------------------------------------------
Consistency Checker Result for GPC:  SUCCESS
-------------------------------------------------------------
Interface properties checks (Module 2):
NATIVE_VLAN: PASSED
FEX_STATUS: PASSED
SPEED: PASSED
FLOW_CONTROL: PASSED
MTU: PASSED
-------------------------------------------------------------
Module 2: PASSED.
```

```
-------------------------------------------------------------
Interface properties checks (Module 4):
NATIVE_VLAN: PASSED
FEX_STATUS: PASSED
SPEED: PASSED
FLOW_CONTROL: PASSED
MTU: PASSED
-------------------------------------------------------------
Module 4: PASSED.
-------------------------------------------------------------
Link State Checks :
-------------------------------------------------------------
Module 2: PASSED
-------------------------------------------------------------
Link State Checks :
-------------------------------------------------------------
Module 4: PASSED
-------------------------------------------------------------
-------------------------------------------------------------
Consistency Checker Result for RPC: SUCCESS
-------------------------------------------------------------
-------------------------------------------------------------
Consistency Checker Result for STP (VLAN CBL): SUCCESS
-------------------------------------------------------------
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 2: SUCCESS
==============================================================
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 4: SUCCESS
==============================================================
PSS CONSISTENCY CHECK RESULT FOR ELTM: FAILURE
---------------------------------------------------------------
 ATTRIBUTE NAME     : ELTM INTERFACE PSS
 INCONSISTENT DATA  : intf Vlan4040 (0x9010fc8)
Please collect the tech-support for eltm detail for more details.
==============================================================
PSS CONSISTENCY CHECK RESULT FOR ETHPM: SUCCESS
---------------------------------------------------------------
No inconsistency detected in ethpm persistent, runtime and shared data.
==============================================================
PSS CONSISTENCY CHECK RESULT FOR STP: SUCCESS
---------------------------------------------------------------
No inconsistency detected in STP CBL data
==============================================================
PSS CONSISTENCY CHECK RESULT FOR VLAN_MGR: SUCCESS
---------------------------------------------------------------
No inconsistency detected in vlan_mgr persistent, runtime and shared data.
==============================================================
PSS CONSISTENCY CHECK RESULT FOR vPC MGR: SUCCESS
---------------------------------------------------------------
No inconsistency detected in vPC persistent, runtime and shared data.
==============================================================

Consistency-checker took 161 secs.
switch#
```

Example: Show Consistency Checker Interface Properties Output

```
switch# show consistency-checker interface-properties

Interface properties checks (Module 4):
NATIVE_VLAN: PASSED
FEX_STATUS: PASSED
SPEED: PASSED
FLOW_CONTROL: PASSED
```

```
MTU: PASSED
-------------------------------------------------------------
Module 4: PASSED.
-------------------------------------------------------------

switch#
```

Example: Show Consistency Checker Link State Output

```
switch# show consistency-checker link-state


Link State Checks :
-------------------------------------------------------------
Module 4: PASSED
-------------------------------------------------------------
switch#
```

Example: Show Consistency Checker L2Unicast Output

```
switch# show consistency-checker l2unicast 1
  Consistency Checker Status: Success


switch# show consistency-checker l2unicast 1
Missing entries in the MAC Table
   VLAN     MAC Address      Type      age     Secure NTFY    Ports
---------+-----------------+--------+---------+------+----+------------------
1201     64a0.e741.2bc1    dynamic     ~~~      F    F  Po100
Extra entries in the MAC Table
   VLAN     MAC Address      Type      age     Secure NTFY    Ports
---------+-----------------+--------+---------+------+----+------------------
  1201     64a0.e741.2bc1    dynamic     ~~~      F    F  Po100
  1202     64a0.e741.2bc1    dynamic     ~~~      F    F  Po100
Discrepant entries in the MAC Table
   VLAN     MAC Address      Type      age     Secure NTFY    Ports
---------+-----------------+--------+---------+------+----+------------------
* 2913     0000.3f80.a6e2    static       -       T    T  Eth153/1/17
* 2914     0000.3f80.a6e4    static       -       T    T  Eth153/1/18
* 2915     0000.3f80.a6e6    static       -       T    T  Eth15

Consistency-Checker: Failure
```

Example: Show Consistency Checker L2Multicast Output

```
switch# show consistency-checker l2mcast all

  Module 10 : Success
  Module 1 : Success
  Module 3 : Success
  Module 2 : Success
  Module 4 : Not Supported
  Module 7 : Not Supported
  Module 9 : Success
  Module 8 : Success
  Consistency Checker Status: Success
```

Example: Show Consistency Checker Spanning-Tree Output

```
switch# show consistency-checker stp-state
-------------------------------------------------------------
Consistency Checker Result for STP (VLAN CBL): SUCCESS
-------------------------------------------------------------
```

```
switch# show consistency-checker stp-state
Consistency Checker Result for STP (VLAN CBL): FAILED
STP/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
PIXM/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35
```

Example: Show Consistency Checker PSS Output

```
switch# show consistency-checker pss
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 2: SUCCESS
==============================================================
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 4: SUCCESS
==============================================================
PSS CONSISTENCY CHECK RESULT FOR ELTM: FAILURE
---------------------------------------------------------------
 ATTRIBUTE NAME     : ELTM INTERFACE PSS
 INCONSISTENT DATA  : intf Vlan4040 (0x9010fc8)
 ATTRIBUTE NAME     : ELTM INTERFACE PSS
 INCONSISTENT DATA  : intf port-channel200 (0x160000c7)
Please collect the tech-support for eltm detail for more details.
==============================================================
PSS CONSISTENCY CHECK RESULT FOR ETHPM: SUCCESS
---------------------------------------------------------------
No inconsistency detected in ethpm persistent, runtime and shared data.
==============================================================
PSS CONSISTENCY CHECK RESULT FOR STP: SUCCESS
---------------------------------------------------------------
No inconsistency detected in STP CBL data
==============================================================
PSS CONSISTENCY CHECK RESULT FOR VLAN_MGR: SUCCESS
---------------------------------------------------------------
No inconsistency detected in vlan_mgr persistent, runtime and shared data.
==============================================================
PSS CONSISTENCY CHECK RESULT FOR vPC MGR: SUCCESS
---------------------------------------------------------------
No inconsistency detected in vPC persistent, runtime and shared data.
==============================================================
```

Example: Show Consistency Checker PSS Output

```
switch# show consistency-checker l3-interface port-channel 5
Consistency Checker Result for Interface: port-channel5  :  Success
```

```
switch# show consistency-checker l3-interface port-channel 5
Consistency Checker Result for Interface: port-channel5  :  Failure
Total Errors Found       :  1
Found error on slot 9 Intf: port-channel5 (0x16000004) : SDB error(1)
Errors detected.  Please collect the output of 'show tech-support eltm detail'.
```

Example: Show Consistency Checker FabricPath Output

```
switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC:  SUCCESS
```

```
switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC : FAILED
    gpc1:1005 not found in PIXM DB
    gpc1:1008 not found in PIXM DB
```

Example: Show Consistency Checker Proxy RPC Output

```
switch# show consistency-checker proxy vl3-membership
Consistency Checker Result for Proxy VL3: SUCCESS
```

```
switch# show consistency-checker proxy vl3-membership
Consistency Checker Result for Proxy VL3: FAILED
MCM VL3 members: Eth1/3  Eth1/4
PIXM VL3 members: Eth1/3
```

Output Examples for Consistency Checker Components – Cisco NX-OS Release 8.0(1)

Example: Link State Output

This example shows a link state output:

```
switch# show consistency-checker link-state
Link State Checks:
Consistency Check: FAILED
Inconsistencies found for following interfaces:
Ethernet1/12 hw_link_state(0) sw_link_state(1)
```

Example: STP Output

This example shows an STP output when the Consistency Checker result for STP passed:

```
switch# show consistency-checker stp-state
Consistency Checker Result for STP (VLAN CBL): SUCCESS
```

This example shows an STP output when the Consistency Checker result for STP failed:

```
switch# show consistency-checker stp-state
Consistency Checker Result for STP (VLAN CBL): FAILED

STP/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
PIXM/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35

Please collect the output of 'show tech-support spanning-tree'.
```

Example: PIXM (FabricPath) Output

This example shows a PIXM output when the Consistency Checker result for PIXM passed:

```
switch# show consistency-checker fabricpath ftag-state
Consistency Checker Result for Ftag CBL: SUCCESS
```

```
switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC: SUCCESS
```

These examples show PIXM outputs when the Consistency Checker result for PIXM failed:

```
switch# show consistency-checker fabricpath ftag-state
Consistency Checker Result for Ftag CBL: FAILED
PIXM/HW FTag CBL mismatch (port Eth3/9):
    INGRESS FORWARDING: (PIXM) 1-2, (HW) 1-2,30-35
    EGRESS FORWARDING: (PIXM) 1-2, (HW) 1-2,30-35
```

```
switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC : FAILED
  gpc3:22
  PIXM members:   Eth2/2
  MCM members:  Eth2/2 Eth2/3
```

```
switch# show consistency-checker proxy rpc-membership
Consistency Checker Result for RPC: FAILED
PIXM vl3 members: Eth4/3
MCM vl3 members:  Eth4/1 Eth4/10 Eth4/17 Eth4/18 Eth4/2 Eth4/25   Eth4/26 Eth4/9 Eth9/1
Eth9/10 Eth9/17 Eth9/18 Eth9/2 Eth9/25 Eth9/26 Eth9/9
```

Example: L2MCAST Output

This example shows a L2MCAST output when the Consistency Checker result for L2MCAST passed:

```
switch(config)# show consistency-checker l2mcast 500 239.2.3.5
Consistency Checker Status: Passed
```

These examples show L2MCAST outputs when the Consistency Checker result for L2MCAST failed:

```
switch(config)# show consistency-checker l2mcast 500 239.2.3.5
Consistency Checker Status: Failed
Inconsistency found in Layer 2 Multicast NextHop
Detailed logs can be found with "show consistency-checker l2mcast vlan group [source]” with
 detail keyword.
```

```
switch(config)# show consistency-checker l2mcast 500 239.2.3.5 detail
Consistency Checker Status: Failed

--------------------------------------------
Route: ('500', '10.120.33.63', '239.2.3.5')
--------------------------------------------
B - Baseline
C - Route and Next-Hop Consistent
I - Next-Hop Inconsistent
M - Missing Route
IGMP: ( B ) set([u'Eth7/9/3'])
M2RIB: ( C ) set([u'Eth7/9/3'])
MFDM: ( C ) 0x7be4
PIXM: ( I ) set(['Eth7/9/3’, ‘Eth7/9/2’])
```

Example: Interface Properties Output

This example shows an interface properties output:

```
switch# show consistency-checker interface-properties
Interface properties checks :
Consistency Check (native_vlan) : PASSED
Consistency Check (fex_status) : PASSED
Consistency Check (speed) : FAILED
Inconsistencies found for following interfaces:
Ethernet1/12 hw_speed(10000) sw_speed(1000)
Consistency Check (flow_control) : PASSED
Please collect the output of 'show tech-support ethpm’
```

Example: L3 Interface Properties Output

This example shows an L3 interface properties output when the Consistency Checker result for L3 interface passed:

```
switch# show consistency-checker l3-interface ethernet 3/6
Consistency Checker Result for Interface:Ethernet3/6 : Success
```

This example shows an L3 interface properties output when the Consistency Checker result for L3 interface failed:

```
switch# show consistency-checker l3-interface ethernet 3/6
Consistency Checker Result for Interface:Ethernet3/6 : Failure
Total Errors Found : 1
Found error on slot 3 Intf:Ethernet3/6 (0x1a105000) : ELM error(19)
Errors detected. Please collect the output of 'show tech-support eltm detail'.
```

Fault Management System

The Fault Management System is used to enhance Cisco NX-OS serviceability by providing an efficient means to capture data that is relevant and adequate to debug the issues being reported at the earliest possible time, without any manual intervention. If all the nodes are down, the packets get routed automatically.

The Fault Management System provides two main benefits in enhancing Cisco NX-OS serviceability:

- **Trigger-based auto capture**—The Fault Management System provides a set of programmable hooks that can be inserted at various predefined (failure) points in such a way that the relevant data is captured automatically whenever a trigger is detected. The data collected by this system includes ASCII tech support, binary tech support, global message and transaction service data, various process-specific details, and specific **show** commands. This system is designed to capture data in the least intrusive way possible.

- **Message and transaction service statistics**—The Fault Management System provides an extension to the message and transaction service infrastructure (mtstrack) library that collects per-process and global message and transaction service statistics. The statistical results can be displayed and analyzed, as required. Message and transaction service statistics (mtstrack) feature is incorporated with the Auto Capture feature to work as an Auto Capture trigger. Using the Auto Capture trigger, any message and transaction service leak in the system can be detected and the **show tech-support** command output can be captured automatically. As with the message and transaction service statistics Auto Capture trigger, trigger points can be identified on other infra components and auto triggers can be added.

Programmability in the Fault Management System

This feature provides a flexible infra and provides functionalities to tweak the behavior of the system to meet the requirements of every Cisco NX-OS process.

The behavior of the system can be programmed using a YAML file. A system default YAML file is present; this can be overwritten with a custom YAML file. When a custom YAML file is used, programming is performed incrementally over the system YAML file.

Note The custom YAML file name must be *fault-mgmt.yaml* in order to enable the file to overwrite the existing YAML file.

This example shows the contents of a YAML file:

```
applications:
 vlan:
         ts_name: vlan
         group_ts_name: "private-vlan,ethpm"
         max_msg_timeout: 30
 ethpm:
        ts_name: ethpm
        group_ts_name: "vlan,lim"
```

```
       max_msg_timeout: 30
       auto_trigger_disable_eve_seq_failure: 1
"private-vlan":
       ts_name: "private-vlan"
       group_ts_name: "ethpm,vlan,stp"
       max_msg_timeout: 30
"eltm detail":
       ts_name: "eltm detail"
       group_ts_name: "vlan,vni"
       max_msg_timeout: 30
"vpc":
      max_msg_timeout: 30
      auto_trigger_disable_eve_seq_failure: 1
```

The following table provides information about semantics used in the YAML file:

Table 2: YAML Semantics

Component	Description
ts_name	Specifies the technical support name for the given application.
group_ts_name	Specifies the names of the applications in the group of a given application.
auto_trigger_disable_mts_timeout	Disables message and transaction service leak detection.
max_msg_timeout	Specifies the message and transaction service leak detection time, in minutes.
auto_trigger_disable_eve_seq_failure	Disables auto trigger on event sequence failure.
auto_trigger_syslog_severity: *severity level*	Specifies syslog severity for the auto capture trigger. Severity level range is from 1 to 7. We do not recommend a severity level above 3.

Adding a Custom YAML File

Procedure

Step 1 Place the YAML file in the **bootflash:scripts/** directory.

Step 2 Use the **fault-management yaml reconfigure** command to overwrite the default YAML file.

Note The custom YAML file name must be *fault-mgmt.yaml* in order to enable the file to overwrite the existing YAML file.

Configuring the Auto Capture Feature

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **fault-management auto-capture**	**Note** The Auto Capture feature is enabled by default. If the Auto Capture feature is disabled, use this command to enable the feature. Use the following information to perform additional configurations in the Auto Capture feature: • Use the [**no**] **fault-management auto-capture** command to disable this feature. • Use the **dir bootflash:fault-management-logs/** command to list the auto captured files. • Use the **clear fault-management logs** [**active** | **standby** | **all**] command to clear the auto captured files.

Configuring the MTS Statistics Feature

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system statistics mts sap** *sap-number* | **all** [**module** *module-number*]	Enables the Message and Transaction Service Statistics feature. **Note** The Message and Transaction Service Statistics feature is enabled by default. Use the following commands to perform additionally configurations in the Message and Transaction Service Statistics feature: • Use the [**no**] **system statistics mts sap** *sap-number* | **all** [**module** *module-number*] command to disable this feature.

	Command or Action	Purpose
		• Use the **show system statistics mts sap** {*sap-number* | **all**} {**brief** | **module** | **receive** | **transmit**} {**us** | **ms** | **detail**} [**sort** {**ascending** | **descending**} **by** {**last-time** | **max-time** | **avg-time** | **count**} command to display Message and Transaction Service Statistics. **Caution** We recommended that you do not use the **all** keyword for service access points (SAPs) because it retrieves data from all the components, which may, in turn results in a long output. Instead, use the *sap-num* argument to retrieve data from a specific component. • Use the **clear statistics mts sap** {**all** | *sap-number*} [**module** *module-number*] command to reset the Message and Transaction Service Statistics.

Configuration Examples for Fault Management System

Example: Enabling the Auto Capture Feature

This example shows how to enable the Auto Capture feature:

```
switch# configure terminal
switch(config)# fault-management auto-capture
```

Example: Enabling the Message and Transaction Service Statistics Feature

This example shows how to enable the Message and Transaction Service Statistics feature:

```
switch# configure terminal
switch(config)# system statistics mts sap all
```

Example: Clearing the Fault-Management Logs

This example shows how to clear the fault-management logs:

```
switch# configure terminal
switch(config)# clear fault-management logs all
```

Example: Programming the System YAML File

This example shows how to program the system YAML file incrementally:

```
switch# configure terminal
switch(config)# fault-management yaml reconfigure
```

Manageability

This section describes the manageability features in the Cisco NX-OS software.

Simple Network Management Protocol

The Cisco NX-OS software is compliant with Simple Network Management Protocol (SNMP) version 1, version 2, and version 3. A large number of MIBs is supported. For more information about SNMP, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Configuration Verification and Rollback

The Cisco NX-OS software allows you to verify the consistency of a configuration and the availability of necessary hardware resources prior to committing the configuration. You can preconfigure a device and apply the verified configuration at a later time. Configurations also include checkpoints that allow you to roll back to a known good configuration as needed. For more information about rollbacks, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Role-Based Access Control

With role-based access control (RBAC), you can limit access to device operations by assigning roles to users. You can customize access and restrict it to the users who require it. For more information about RBAC, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*.

Connectivity Management Processor

The Cisco NX-OS software supports the use of a Connectivity Management Processor (CMP) for remote platform management. The CMP provides an out-of-band access channel to the Cisco NX-OS console. For more information about CMP, see the *Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*.

Cisco NX-OS Device Configuration Methods

You can configure devices using the CLI from a Secure Shell (SSH) session or a Telnet session. SSH provides a secure connection to the device. The CLI configuration guides and command references are organized by feature. For more information, see the Cisco NX-OS configuration guides and the Cisco NX-OS command references. For more information on SSH and Telnet, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*.

You can also configure devices using the XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI. For more information, see the *Cisco Nexus 7000 Series NX-OS Programmability Guide*.

Traffic Routing, Forwarding, and Management

This section describes the traffic routing, forwarding, and management features supported by the Cisco NX-OS software.

Ethernet Switching

The Cisco NX-OS software supports high-density, high-performance Ethernet systems and provides the following Ethernet switching features:

- IEEE 802.1D-2004 Rapid and Multiple Spanning Tree Protocols (802.1w and 802.1s)
- IEEE 802.1Q VLANs and trunks
- 16,000-subscriber VLANs
- IEEE 802.3ad link aggregation
- Private VLANs
- Cross-chassis private VLANs
- Unidirectional Link Detection (UDLD) in aggressive and standard modes

For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 6.x* and the *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide*.

IP Routing

The Cisco NX-OS software supports IP version 4 (IPv4) and IP version 6 (IPv6) and the following routing protocols:

- Open Shortest Path First (OSPF) Protocol Versions 2 (IPv4) and 3 (IPv6)
- Intermediate System-to-Intermediate System (IS-IS) Protocol
- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Routing Information Protocol Version 2 (RIPv2)

The Cisco NX-OS software implementations of these protocols are fully compliant with the latest standards and include 4-byte autonomous system numbers (ASNs) and incremental shortest path first (SPF). All unicast protocols support Non-Stop Forwarding Graceful Restart (NSF-GR). All protocols support all interface types, including Ethernet interfaces, VLAN interfaces, subinterfaces, port channels, tunnel interfaces, and loopback interfaces.

For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide*.

IP Services

The following IP services are available in the Cisco NX-OS software:

- Virtual Routing and Forwarding (VRF)
- Dynamic Host Configuration Protocol (DHCP) Helper
- Hot-Standby Routing Protocol (HSRP)
- Gateway Load Balancing Protocol (GLBP)
- Enhanced Object Tracking
- Policy-Based Routing (PBR)
- Unicast Graceful Restart for all protocols in IPv4 Unicast Graceful Restart for OPSFv3 in IPv6

For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide*.

IP Multicast

The Cisco NX-OS software includes the following multicast protocols and functions:

- Protocol Independent Multicast (PIM) Version 2 (PIMv2)
- Source Specific Multicast (SSM)
- PIM sparse mode (Any-Source Multicast [ASM] for IPv4 and IPv6)

Note The Cisco NX-OS software does not support PIM dense mode.

- Bidirectional Protocol Independent Multicast (Bidir PIM)
- Anycast rendezvous point (Anycast-RP)
- Multicast NSF for IPv4 and IPv6
- RP-Discovery using bootstrap router (BSR) (Auto-RP and static)
- Internet Group Management Protocol (IGMP) Versions 1, 2, and 3 router role
- IGMPv2 host mode
- IGMP snooping
- Multicast Listener Discovery (MLD) Protocol Version 2 (for IPv6)
- Multicast Source Discovery Protocol (MSDP) (for IPv4 only)

For more information, see the *Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide*.

Quality of Service

The Cisco NX-OS software supports quality of service (QoS) functions for classification, marking, queuing, policing, and scheduling. Modular QoS CLI (MQC) supports all QoS features. You can use MQC to provide uniform configurations across various Cisco platforms. For more information, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide.*

Network Security

This section describes the network security features support by the Cisco NX-OS software.

Cisco TrustSec

Cisco TrustSec security provides data confidentiality and integrity and supports standard IEEE 802.1AE link-layer cryptography with 128-bit Advanced Encryption Standard (AES) cryptography. Link-layer cryptography guarantees end-to-end data privacy while allowing the insertion of security service devices along the encrypted path. Cisco TrustSec uses security group access control lists (SGACLs), which are based on security group tags instead of IP addresses. SGACLs enable policies that are more concise and easier to manage due to their topology independence. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide.*

Additional Network Security Features

In addition to Cisco TrustSec, the Cisco NX-OS software includes the following security features:

- Data path intrusion detection system (IDS) for protocol conformance checks
- Control Plane Policing (CoPP)
- Message-digest algorithm 5 (MD5) routing protocol authentication
- Cisco-integrated security features, including Dynamic Address Resolution Protocol (ARP) inspection (DAI), DHCP snooping, and IP Source Guard
- Authentication, authorization, and accounting (AAA)
- RADIUS and TACACS+
- SSH Protocol Version 2
- SNMPv3
- Port security
- IEEE 802.1X authentication
- Layer 2 Cisco Network Admission Control (NAC) LAN port IP
- Policies based on MAC and IPv4 addresses supported by named ACLs (port-based ACLs [PACLs], VLAN-based ACLs [VACLs], and router-based ACLs [RACLs])
- Traffic storm control (unicast, multicast, and broadcast)

- Unicast Reverse Path Forwarding (Unicast RPF)

For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*.

Supported Standards

This table lists the IEEE compliance standards.

Table 3: IEEE Compliance Standards

Standard	Description
802.1D	MAC Bridges
802.1s	Multiple Spanning Tree Protocol
802.1w	Rapid Spanning Tree Protocol
802.1AE	MAC Security (link layer cryptography)
802.3ad	Link aggregation with LACP
802.3ab	1000BASE-T (10/100/1000 Ethernet over copper)
802.3ae	10-Gigabit Ethernet
802.1Q	VLAN Tagging
802.1p	Class of Service Tagging for Ethernet frames
802.1X	Port-based network access control

This table lists the RFC compliance standards.

Table 4: RFC Compliance Standards

Standard	Description
BGP	
RFC 1997	BGP Communities Attribute
RFC 2385	Protection of BGP Sessions via the TCP MD5 Signature Option
RFC 2439	BGP Route flap damping
RFC 2519	A Framework for Inter-Domain Route Aggregation
RFC 2858	Multiprotocol Extensions for BGP-4

Standard	Description
RFC 3065	Autonomous System Confederations for BGP
RFC 3392	Capabilities Advertisement with BGP-4
RFC 4271	BGP version 4
RFC 4273	BGP4 MIB - Definitions of Managed Objects for BGP-4
RFC 4456	BGP Route reflection
RFC 4486	Subcodes for BGP cease notification message
RFC 4724	Graceful Restart Mechanism for BGP
RFC 4893	BGP Support for Four-octet AS Number Space
ietf-draft	Bestpath transition avoidance (draft-ietf-idr-avoid-transition-05.txt)
ietf-draft	Peer table objects (draft-ietf-idr-bgp4-mib-15.txt)
ietf-draft	Dynamic Capability (draft-ietf-idr-dynamic-cap-03.txt)
OSPF	
RFC 2370	OSPF Opaque LSA Option
RFC 2328	OSPF Version 2
RFC 2740	OSPF for IPv6 (OSPF version 3)
RFC 3101	OSPF Not-So-Stubby-Area (NSSA) Option
RFC 3137	OSPF Stub Router Advertisement
RFC 3509	Alternative Implementations of OSPF Area Border Routers
RFC 3623	Graceful OSPF Restart
RFC 4750	OSPF Version 2 MIB
RIP	
RFC 1724	RIPv2 MIB extension

Standard	Description
RFC 2082	RIPv2 MD5 Authentication
RFC 2453	RIP Version 2
IS-IS	
RFC 1142 (OSI 10589)	OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol
RFC 1195	Use of OSI IS-IS for routing in TCP/IP and dual environment
RFC 2763	Dynamic Hostname Exchange Mechanism for IS-IS
RFC 2966	Domain-wide Prefix Distribution with Two-Level IS-IS
RFC 2973	IS-IS Mesh Groups
RFC 3277	IS-IS Transient Blackhole Avoidance
RFC 3373	Three-Way Handshake for IS-IS Point-to-Point Adjacencies
RFC 3567	IS-IS Cryptographic Authentication
RFC 3847	Restart Signaling for IS-IS
ietf-draft	Internet Draft Point-to-point operation over LAN in link-state routing protocols (draft-ietf-isis-igp-p2p-over-lan-06.txt)
IP Services	
RFC 768	UDP
RFC 783	TFTP
RFC 791	IP
RFC 792	ICMP
RFC 793	TCP
RFC 826	ARP
RFC 854	Telnet
RFC 959	FTP

Standard	Description
RFC 1027	Proxy ARP
RFC 1305	NTP v3
RFC 1519	CIDR
RFC 1542	BootP relay
RFC 1591	DNS client
RFC 1812	IPv4 routers
RFC 2131	DHCP Helper
RFC 2338	VRRP
RFC 2784	Generic Routing Encapsulation (GRE)
IP-Multicast	
RFC 2236	Internet Group Management Protocol, Version 2
RFC 2710	Multicast Listener Discovery (MLD) for IPv6
RFC 3376	Internet Group Management Protocol, Version 3
RFC 3446	Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)
RFC 3569	An Overview of Source-Specific Multicast (SSM)
RFC 3618	Multicast Source Discovery Protocol (MSDP)
RFC 3810	Multicast Listener Discovery Version 2 (MLDv2) for IPv6
RFC 4601	ASM - Sparse Mode (PIM-SM): Protocol Specification (Revised)
RFC 4607	Source-Specific Multicast for IP
RFC 4610	Anycast-RP Using Protocol Independent Multicast (PIM)

Standard	Description
ietf-draft	Mtrace server functionality, to process mtrace-requests, draft-ietf-idmr-traceroute-ipm-07.txt
ietf-draft	Bi-directional Protocol Independent Multicast (BIDIR-PIM), draft-ietf-pim-bidir-09.txt

CHAPTER 3

Using the Cisco NX-OS Setup Utility

This chapter describes how to use the Cisco NX-OS setup utility.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Prerequisites for the Setup Utility

The setup utility has the following prerequisites:

- Have a password strategy for your network environment.
- Connect the console port on the supervisor module to the network. If you have dual supervisor modules, connect the console ports on both supervisor modules to the network.
- Connect the Ethernet management port on the supervisor module to the network. If you have dual supervisor modules, connect the Ethernet management ports on both supervisor modules to the network.

Information About the Cisco NX-OS Setup Utility

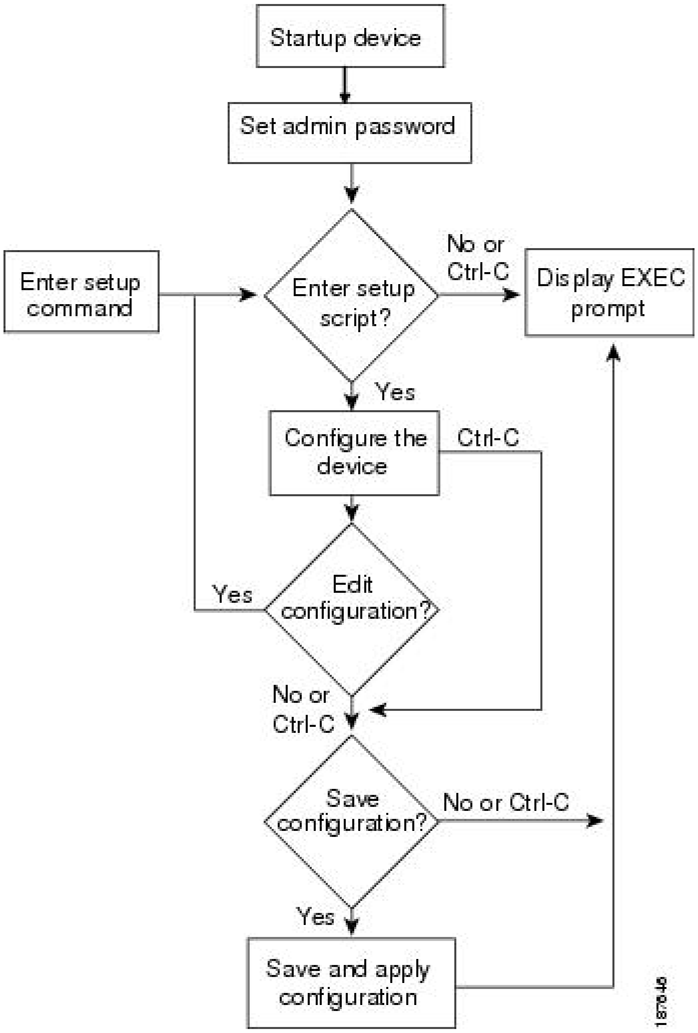
The Cisco NX-OS setup utility is an interactive command-line interface (CLI) mode that guides you through a basic (also called a startup) configuration of the system. The setup utility allows you to configure only enough connectivity for system management.

The setup utility allows you to build an initial configuration file using the System Configuration Dialog. The setup starts automatically when a device has no configuration file in NVRAM. The dialog guides you through initial configuration. After the file is created, you can use the CLI to perform additional configuration.

You can press **Ctrl-C** at any prompt to skip the remaining configuration options and proceed with what you have configured up to that point, except for the administrator password. If you want to skip answers to any questions, press **Enter**. If a default answer is not available (for example, the device hostname), the device uses what was previously configured and skips to the next question.

Figure 2: Setup Script Flow

This figure shows how to enter and exit the setup script.



You use the setup utility mainly for configuring the system initially, when no configuration is present. However, you can use the setup utility at any time for basic device configuration. The setup utility keeps the configured values when you skip steps in the script. For example, if you have already configured the mgmt0 interface, the setup utility does not change that configuration if you skip that step. However, if there is a default value for the step, the setup utility changes to the configuration using that default, not the configured value. Be sure to carefully check the configuration changes before you save the configuration.

Note Be sure to configure the IPv4 route, the default network IPv4 address, and the default gateway IPv4 address to enable SNMP access. If you enable IPv4 routing, the device uses the IPv4 route and the default network IPv4 address. If IPv4 routing is disabled, the device uses the default gateway IPv4 address.

Note The setup script only supports IPv4.

Setting Up Your Cisco NX-OS Device

To configure basic management of the Cisco NX-OS device using the setup utility, follow these steps:

Procedure

Step 1 Power on the device.

Step 2 Enable or disable password-strength checking.

A strong password has the following characteristics:

- At least eight characters long
- Does not contain many consecutive characters (such as "abcd")
- Does not contain many repeating characters (such as "aaabbb")
- Does not contain dictionary words
- Does not contain proper names
- Contains both uppercase and lowercase characters
- Contains numbers

Example:

```
         ---- System Admin Account Setup ----

Do you want to enforce secure password standard (yes/no) [y]: y
```

Step 3 Enter the new password for the administrator.

Note If a password is trivial (such as a short, easy-to-decipher password), your password configuration is rejected. Passwords are case sensitive. Be sure to configure a strong password that has at least eight characters, both uppercase and lowercase letters, and numbers.

Example:

```
Enter the password for "admin": <password>

Confirm the password for "admin": <password>
```

```
---- Basic System Configuration Dialog VDC: 1 ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

Please register Cisco Nexus7000 Family devices promptly with your
supplier. Failure to register may affect response times for initial
service calls. Nexus7000 devices must be registered to receive
entitled support services.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.
```

Step 4 Enter the setup mode by entering **yes**.

Example:

```
Would you like to enter the basic configuration dialog (yes/no): yes
```

Step 5 Create additional accounts by entering **yes** (**no** is the default).

Example:

```
  Create another login account (yes/no) [n]:yes
```

a) Enter the user login ID.

Example:

```
Enter the User login Id : user_login
```

Caution Usernames must begin with an alphanumeric character and can contain only these special characters: (+ = . _ \ -). The # and ! symbols are not supported. If the username contains characters that are not allowed, the specified user is unable to log in.

b) Enter the user password.

Example:

```
Enter the password for "user1": user_password
Confirm the password for "user1": user_password
```

c) Enter the default user role.

Example:

```
Enter the user role (network-operator|network-admin|vdc-operator|vdc-admin)
[network-operator]: default_user_role
```

For information on the default user roles, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

Step 6 Configure an SNMP community string by entering **yes**.

Example:

```
Configure read-only SNMP community string (yes/no) [n]: yes
SNMP community string : snmp_community_string
```

For information on SNMP, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Step 7 Enter a name for the device (the default name is switch).

Example:

```
Enter the switch name: switch_name
```

Step 8 Configure out-of-band management by entering **yes**. You can then enter the mgmt0 IPv4 address and subnet mask.

Note You can only configure IPv4 address in the setup utility. For information on configuring IPv6, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x*.

Example:

```
Continue with Out-of-band (mgmt0) management configuration? [yes/no]: yes
Mgmt0 IPv4 address: mgmt0_ip_address
Mgmt0 IPv4 netmask: mgmt0_subnet_mask
```

Step 9 Configure the IPv4 default gateway (recommended) by entering **yes**. You can then enter its IP address.

Example:

```
Configure the default-gateway: (yes/no) [y]: yes
IPv4 address of the default-gateway: default_gateway
```

Step 10 Configure advanced IP options such as the static routes, default network, DNS, and domain name by entering **yes**.

Example:

```
Configure Advanced IP options (yes/no)? [n]: yes
```

Step 11 Configure a static route (recommended) by entering **yes**. You can then enter its destination prefix, destination prefix mask, and next hop IP address.

Example:

```
Configure static route: (yes/no) [y]: yes
Destination prefix: dest_prefix
Destination prefix mask: dest_mask
Next hop ip address: next_hop_address
```

Step 12 Configure the default network (recommended) by entering **yes**. You can then enter its IPv4 address.

Note The default network IPv4 address is the same as the destination prefix in the static route configuration.

Example:

```
Configure the default network: (yes/no) [y]: yes
Default network IP address [dest_prefix]: dest_prefix
```

Step 13 Configure the DNS IPv4 address by entering **`yes`**. You can then enter the address.

Example:

```
Configure the DNS IP address? (yes/no) [y]: yes
DNS IP address: ipv4_address
```

Step 14 Configure the default domain name by entering **`yes`**. You can then enter the name.

Example:

```
Configure the DNS IP address? (yes/no) [y]: yes
DNS IP address: ipv4_address
```

Step 15 Enable the Telnet service by entering **`yes`**.

Example:

```
Enable the telnet service? (yes/no) [y]: yes
```

Step 16 Enable the SSH service by entering **`yes`**. You can then enter the key type and number of key bits. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

Example:

```
Enable the ssh service? (yes/no) [y]: yes
Type of ssh key you would like to generate (dsa/rsa) : key_type
Number of  key bits <768-2048> : number_of_bits
```

Step 17 Configure the NTP server by entering **`yes`**. You can then enter its IP address. For more information, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Example:

```
Configure NTP server? (yes/no) [n]: yes
NTP server IP address: ntp_server_IP_address
```

Step 18 Specify a default interface layer (L2 or L3).

Example:

```
Configure default interface layer (L3/L2) [L3]: interface_layer
```

Step 19 Enter the default switchport interface state (shutdown or no shutdown). A shutdown interface is in an administratively down state. For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 6.x*.

Example:

```
Configure default switchport interface state (shut/noshut) [shut]: default_state
```

Step 20 Enter the best practices profile for control plane policing (CoPP). For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

Example:

```
Configure best practices CoPP profile (strict/moderate/lenient/none) [strict]: policy
```

Step 21 Configure CMP for the current supervisor, and then enter the IP address, netmask, and default gateway IP by entering **yes**. For more information, see the *Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*.

Example:

```
Configure CMP processor on current sup (slot 5)? (yes/no) [y]: yes
cmp-mgmt IPv4 address : IP_address
cmp-mgmt IPv4 netmask : net_mask
IPv4 address of the default gateway : default_gateway
```

Step 22 Configure CMP for the redundant supervisor by entering **yes**. You can then enter the IP address, netmask, and default gateway IP.

Example:

```
Configure CMP processor on standby sup (slot 5)? (yes/no) [y]: yes
cmp-mgmt IPv4 address : IP_address
cmp-mgmt IPv4 netmask : net_mask
IPv4 address of the default gateway : default_gateway
```

The system now summarizes the complete configuration and asks if you want to edit it.

Step 23 Continue to the next step by entering **no**. If you enter **yes**, the setup utility returns to the beginning of the setup and repeats each step.

Example:

```
Would you like to edit the configuration? (yes/no) [y]: yes
```

Step 24 Use and save this configuration by entering **yes**. If you do not save the configuration at this point, none of your changes are part of the configuration the next time the device reboots. Enter **yes** to save the new configuration. This step ensures that the boot variables for the kickstart and system images are also automatically configured.

Example:

```
Use this configuration and save it? (yes/no) [y]: yes
```

Caution If you do not save the configuration at this point, none of your changes are part of the configuration the next time that the device reboots. Enter **yes** to save the new configuration to ensure that the boot variables for the kickstart and system images are also automatically configured.

Additional References for the Setup Utility

This section includes additional information related to using the setup utility.

Related Documents for the Setup Utility

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*
CMP	*Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*
SSH and Telnet	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*
User roles	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*
IPv4 and IPv6	*Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 6.x*
SNMP and NTP	*Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.xCisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*

CHAPTER 4

Configure Replace

This chapter describes how to configure the Configure Replace feature.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Information About Configure Replace and Commit-timeout

The configuration replace feature enables you to replace the running configuration of the Cisco Nexus switch with the user provided configuration without reloading the device. The device reload may be required only when a configuration itself requires a reload. The user provided configuration is the running configuration that is received with the Cisco NX-OS device. Unlike **copy file: to running**, the configuration replace feature is not a merge operation. This feature replaces the entire running configuration with a new configuration that is provided by the user. If there is a failure in the configuration replace, the original configuration is restored in the switch.

The commit-timeout feature enables you to rollback to the previous configuration after successfully performing the configuration replace operation. If the commit timer expires, the rollback operation is automatically initiated.

The commit-timeout feature is initiated only if you perform the configuration replace operation with the commit-timeout. The timer value range is between 30-3600 seconds.

Note The **class type queuing***new-class-policy* command in the running configuration causes a config-replace failure in Cisco Nexus 7700 Series platform in Cisco NX-OS Release 8.2(1). It is recommended that you remove the **class type queuing***new-class-policy* command from the running or target configuration to prevent the config-replace failure.

Overview

The configuration replace feature leverages the current rollback infrastructure with operation steps as follows:

- Configuration replace intelligently calculates the difference between the current running-configuration and the user-provided configuration in the Cisco Nexus switch and generates a patch file which is the difference between the two files. You can view this patch file which includes a set of configuration commands.

- Configuration replace applies the configuration commands from the patch file similarly to executing commands.

Note Since the configuration replace feature is atomic, if there are any errors while applying the configuration, it breaks at that point and then restores the switch to the original running configuration.

- The configuration rolls back to or restores the previous running configuration under the following situations:
  - If there is a mismatch in the configuration after the patch file has been applied.
  - If you perform the configuration operation with a commit timeout and the commit timer expires.

- You can view the exact configuration that caused a failure using the **show config-replace log exec** command.

- Restore operations that fail while restoring the switch to the original configuration, are not interrupted. The restore operation continues with the remaining configuration. Use the **show config-replace log exec** command to list the commands that failed during the restore operation.

- If you enter the **configure replace commit** command before the timer expires, the commit timer stops and the switch runs on the user provided configuration that has been applied through the configuration replace feature.

- If the commit timer expires, roll back to the previous configuration is initiated automatically.

The differences between configuration replace and copying a file to the running-configuration are as follows:

Configuration Replace	Copying a file
The **configure replace** *<target-url>* command removes the commands from the current running-configuration that are not present in the replacement file. It also adds commands that need to be added to the current running-configuration.	The **copy** *<source-url>* **running-config** command is a merge operation which preserves all the commands from, both the source file and the current running-configuration. This command does not remove the commands from the current running-configuration that are not present in the source file.
You must use a complete Cisco NX-OS configuration file as the replacement file for the **configure replace** *<target-url>* command.	You can use a partial configuration file as a source file for the **copy** *<source-url>* **running-config** command.

Benefits of Configure Replace

The benefits of configuration replace are:

- You can replace the current running-configuration file with the user-provided configuration file without having to reload the switch or manually undo CLI changes to the running-configuration file. As a result, the system downtime is reduced.
- You can revert to the saved Cisco NX-OS configuration state.
- It simplifies the configuration changes by allowing you to apply a complete configuration file to the device, where only the commands that need to be added or removed are affected. The other service and configurations that are not modified remain untouched.
- If you configure the commit-timeout feature, you can rollback to the previous configuration even when the configuration replace operation has been successful.

Prerequisites for Configure Replace

- You need to provide the valid running-configuration taken from the Nexus device. It should not be a partial configuration.

Guidelines and Limitations of Configure Replace

The CR feature has the following configuration guidelines and limitations:

- Commit timeout feature is added in CR in Cisco NX-OS Release 8.3(1).
- The commit-timeout feature is initiated only if you perform the configuration replace operation with the commit-timeout. The timer value range is between 30-3600 seconds.
- The user configuration file to which you need to replace the running configuration on the switch using CR should be generated from the running-config of the switch after configuring the new commands. The user configuration file should not be manually edited with the CLI commands and the sequence of the configuration commands should not be altered.
- The configuration file must be regenerated whenever there is change in the software version.
- It is recommended not to do any of the configuration changes from any other session when CR is in progress. This is to avoid CR failure.
- CR request is serialized; only after the first request is complete the next request is processed.

- CR does not work if the FEX module is offline.
- CR is not supported on port profiles that are inherited on the switch interfaces.
- CR fails if it contains module-specific configuration and if the module is not online.
- CR is supported only for configure terminal mode and configure maintenance mode commands. Configure profile, configure job and any other modes are not supported. Maintenance mode is supported from Cisco NX-OS Release 8.3(1).
- User configuration file must be show run and not show run vdc-all. Configurations taken in one VDC is not applicable to the other VDC.
- CR is not supported on an admin VDC. CR is supported only on the default and non-default VDCs.
- You can perform a parallel CR between different VDCs. For example, user1 can execute CR on VDC1, and user2 can execute CR on VDC2 at the same time, and they will not impact each other.
- To perform parallel CR for more than one VDC; go to the VDC where CR needs to be performed (using the **vdc** <*vdc-name*> command) and execute the **configure replace** <*file-name*> command.
- CR is supported on Supervisor 3 and Fabric Module 3. Starting from Cisco NX-OS 8.4(1), CR is also supported on F4 Series Modules.
- Starting from Cisco NX-OS 8.4(1), CR is supported for breakout interface configurations.

Workflow for Configure Replace operation

The following steps describe the recommended workflow for CR:

1. You can generate a configuration file by first applying the configurations on a Cisco Nexus series switch and then use the copy run file output as the configuration file. This file should be the file where you can make configuration modification as required and use this generated/updated configuration file to perform configuration replace. Make sure the syntax/format for the edited configuration to be same as shown in the running configuration.
2. The configuration file must be regenerated whenever there is change in software version. The CR operation on configuration file generated across software version is not recommended and CR might fail or succeed.
3. You can view and verify the patch file before it gets applied by executing **configure replace** <*file*> **show-patch** command.
4. Run the configuration replace file either using or skipping the commit-timeout feature. Based on your requirements, you can perform one of the following steps:
   - You can run **configure replace** <*file*> **verbose** to see the commands that get executed with CR on console.
   - Run the **configure replace [bootflash/scp/sftp]** <*user-configuration-file*> **verbose** *commit-timeout* <*time*> commands to configure the commit time.
5. Run the **configure replace commit** command to stop the commit timer. This step is necessary if you have run the configuration replace operation with the commit-timeout feature.

6. CR will do pre-check which includes semantic validation of configuration, and in case of error CR exits. The user can use **show config-replace log verify** command to see exact configurations that failed.
7. CR is atomic, in case of failure, the CR exits on the first failure and restores the switch to original configuration. You can use **show config-replace log exec** command to get the error display.
8. Once patch is applied, CR triggers verification where it compares the running-configuration matches with user configuration file, if there is mismatch it restores the switch. You can use **show config-replace verify** command to see mismatched configurations.
9. It is recommended not to modify any configuration through other session when CR in progress.

Performing a Configure Replace

To perform configuration replace, do the following:

Procedure

	Command or Action	Purpose
Step 1	**configure replace** { <*uri_local*> | <*uri_remote*> } [**verbose** | **show-patch**]	Performs configuration replace. If you make the configuration changes through any sessions when configuration replace is in progress, the configuration replace operation fails. If you send a configuration replace request when one configuration request is already in progress, then it gets serialized.
Step 2	**configure replace** [**bootflash**/**scp**/**sftp**] <*user-configuration-file*> **show-patch**	Displays the differences between the running-configuration and the user-provided configuration.
Step 3	**configure replace** [**bootflash**/**scp**/**sftp**] <*user-configuration-file*> **verbose**	Replaces the configuration on the switch with the new user configuration that is provided by the user. Configuration replace is always atomic.
Step 4	(Optional) **configure replace bootflash**/**scp**/**sftp**] <*user-configuration-file*> **verbose** *commit-timeout time*	Configures the commit time in seconds. The timer starts after the configuration replace operation is successfully completed.
Step 5	(Optional) **configure replace** [**commit**]	Stops the commit timer and continues the configuration replace configuration. **Note** This step is applicable only if you have configured the commit-timeout feature.

	Command or Action	Purpose
		Note To rollback to the previous configuration, you must wait for the expiry of the commit timer. Once the timer expires, the switch is automatically rolled back to the previous configuration.
Step 6	(Optional) **configure replace bootflash**/**scp**/**sftp**] <*user-configuration-file*> **non-interactive**	There is no user prompt in maintenance mode. The **yes** user-confirmation is taken by default, and rollback proceeds. The non-interactive option can be used only in the maintenance mode.

Verifying the Configure Replace Operation

The following commands are used to verify the status of the configure replace operation.

Procedure

	Command or Action	Purpose
Step 1	**configure replace [bootflash /scp /sftp]** <*user-configuration-file*>] **show-patch**	Displays the difference between the running-configurations and user-provided configurations.
Step 2	**show config-replace log exec**	Displays a log of all the configurations executed and those that failed. In case of an error, it displays an error message against that configuration.
Step 3	**show config-replace log verify**	Displays the configurations that failed, along with an error message. It does not display configurations that were successful.
Step 4	**show config-replace status**	Displays the status of the configuration replace operations, including in-progress, successful, and failure. If you have configured the commit-timeout feature, the commit and timer status and the commit timeout time remaining is also displayed.

Examples for Configure Replace

See the following configuration examples for configure replace:

- Use the **configure replace bootflash:** <*file*> **show-patch** CLI command to display the difference between the running-configurations and user-provided configurations.

```
switch(config)# configure replace bootflash:<file> show-patch
Collecting Running-Config
Converting to checkpoint file
#Generating Rollback Patch
!!
no role name abc
```

- Use the **configure replace bootflash:** *<file>* **verbose** CLI command to replace the entire running-configuration in the switch with the user-configuration.

```
switch(config)# configure replace bootflash:<file> verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
========================================================
config t
no role name abc
========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

Sample Example with adding of BGP configurations.
 switch(config)# sh run | section bgp
 switch(config)# sh file bootflash:file | section bgp
feature bgp
router bgp 1
    address-family ipv4 unicast
    neighbor 1.1.1.1
switch(config)#
switch(config)# configure replace bootflash:file verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
========================================================
config t
feature bgp
router bgp 1
address-family ipv4 unicast
neighbor 1.1.1.1
========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

switch(config)# sh run | section bgp
feature bgp
router bgp 1
  address-family ipv4 unicast
  neighbor 1.1.1.1

Sample Example with ACL
 switch(config)# configure replace bootflash:run_1.txt
 Collecting Running-Config
```

```
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
========================================================
config t
no ip access-list nexus-50-new-xyz
ip access-list nexus-50-new-xyz-jkl-abc
10 remark Newark
20 permit ip 17.31.5.0/28 any
30 permit ip 17.34.146.193/32 any
40 permit ip 17.128.199.0/27 any
50 permit ip 17.150.128.0/22 any
========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

switch(config)#


switch(config)# show run aclmgr | sec nexus-50-new-xyz-jkl-abc
ip access-list nexus-50-new-xyz-jkl-abc
  10 remark Newark
  20 permit ip 17.31.5.0/28 any
  30 permit ip 17.34.146.193/32 any
  40 permit ip 17.128.199.0/27 any
  50 permit ip 17.150.128.0/22 any
```

- Use the **show config-replace log exec** CLI command to check all the configuration that is executed and failures if any.

```
switch(config)# show config-replace log exec
Operation            : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme               : tmp
Rollback done By     : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
--------------------------------------------------------------------------------

time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
End Time             : Wed, 06:39:47 25 Jan 2017
Rollback Status      : Success

Executing Patch:
----------------
switch#config t
switch#no role name abc
```

- Use the **show config-replace log verify** CLI command to check the failed configuration if any.

```
switch(config)# show config-replace log verify
Operation            : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme               : tmp
Rollback done By     : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
```

```
End Time             : Wed, 06:39:47 25 Jan 2017
Status               : Success

Verification patch contains the following commands:
---------------------------------------------------
!!
! No changes
--------------------------------------------------------------------------------

time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
```

- Use the **show config-replace status** CLI command to check the status of configuration replace.

  ```
  switch(config)# show config-replace status
  Last operation : Rollback to file
  Details:
    Rollback type: atomic replace_tmp_28081
    Start Time: Wed Jan 25 06:39:28 2017
    End Time: Wed Jan 25 06:39:47 2017
    Operation Status: Success
  switch(config)#
  ```

Configure Replace might fail when the manually created configuration has been used instead of the configuration generated from the switch. The reason for possible failures is the potential difference in the default configuration that is not shown in the show running configuration. Refer to the following examples:

If the power redundant command is the default command, it does not get displayed in the default configuration. But it is displayed when you use the **show run all** command. An example is given below.

```
switch# show run all

!Command: show running-config all
!Running configuration last done at: Tue Nov 12 11:07:44 2019
!Time: Tue Nov 12 11:16:09 2019

version 8.4(1) Bios:version 05.39
power redundancy-mode ps-redundant
no hardware module boot-order reverse
no license grace-period
<snip>
hostname brno
```

The power redundant command is not shown in the show running configuration command output. An example is given below.

```
!Command: show running-config
!Running configuration last done at: Tue Nov 12 11:07:44 2019
!Time: Tue Nov 12 11:17:24 2019

version 8.4(1) Bios:version 05.39
hostname brno
```

When the **power redundancy-mode ps-redundant** command is added in the user configuration to be used in configure replace; then the verification/commit might fail. An example is given below.

```
switch# show file bootflash:test

!Command: show running-config
!Running configuration last done at: Tue Nov 12 10:56:49 2019
!Time: Tue Nov 12 11:04:57 2019

version 8.4(1) Bios:version 05.39
```

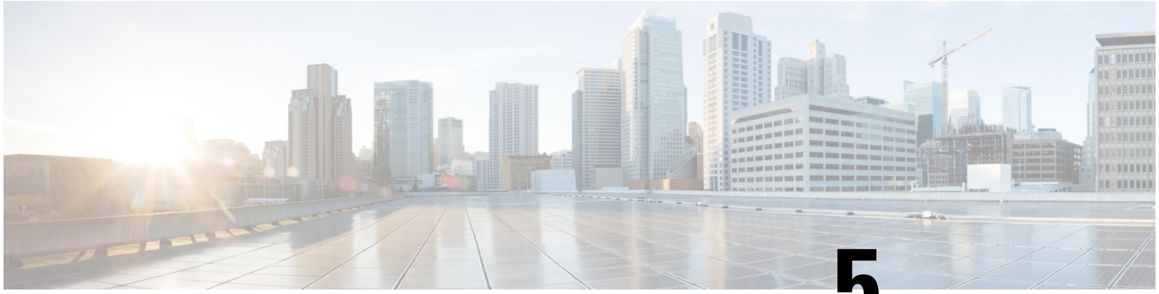
```
power redundancy-mode ps-redundant
hostname brno
```

The **power redundancy-mode ps-redundant** command will not be shown in the show running command output after the configure replace; therefore it will be considered as “missing” and the CR will fail. An example is given below.

```
switch# config replace bootflash:test verify-and-commit

Version match between user file and running configuration.
Pre-check for User config PASSED
ADVISORY: Config Replace operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Config Replace failure.

Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Collecting Running-Config
.Generating Rollback Patch

Validating Patch
Patch validation completed successful
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Rollback Patch
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Patch for verification
Verification failed, Rolling back to previous configuration
Collecting Running-Config
Cleaning up switch-profile buffer
Generating Rollback patch for switch profile
Executing Rollback patch for switch profiles. WARNING - This will change the configuration
 of switch profiles and will also affect any peers if configured
Collecting Running-Config
Generating Rollback Patch
Rollback Patch is Empty
Rolling back to previous configuration is successful

Configure replace failed. Use 'show config-replace log verify' or 'show config-replace log
 exec' to see reasons for failure
Configure replace failed. Use 'show config-replace log verify' or 'show config-replace log
 exec' to see reasons for failure

brno# show config-replace log verify
Operation : Config-replace to user config
Checkpoint file name : .replace_tmp_31849
Scheme : tmp
Cfg-replace done By : agargula
Cfg-replace mode : atomic
Verbose : disabled
Start Time : Tue, 11:20:59 12 Nov 2019
Start Time UTC : Tue, 10:20:59 12 Nov 2019
-------------------------------------------
End Time : Tue, 11:21:28 12 Nov 2019
End Time UTC : Tue, 10:21:28 12 Nov 2019
Status : Failed
Verification patch contains the following commands:
---------------------------------------------------
```

```
!!
Configuration To Be Added Missing in Running-config
===================================================
!
power redundancy-mode ps-redundant
Undo Log
--------------------------------------------------------------------------------
End Time: Tue, 11:21:32 12 Nov 2019
End Time UTC : Tue, 10:21:32 12 Nov 2019
Status : Success
brno#
```

In the above example, CR will consider the default commands that are missing and will therefore fail.



CHAPTER 5

Distributed Packet Tracer

This chapter describes how to configure the Distributed Packet Tracer (DPT) feature using the CLIs.

This chapter contains the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Feature History for Distributed Packet Tracer

This table lists the release history for this feature.

Table 5: Feature History for Distributed Packet Tracer

Feature Name	Releases	Feature Information	
Distributed Packet Tracer (DPT)	8.2(1)	This feature was introduced.	

Information About Distributed Packet Tracer

Distributed Packet Tracer (DPT) is a utility integrated within Cisco Nexus 7000/7700 platforms that can be used to trace the path of the packet through the switch. DPT can be invoked using the command line or remotely using NX-API/JSON/XML and can be configured to match specific traffic flows.

DPT provides information about flows traversing through the switch and the results of forwarding decisions for identified flows such as- forward and drop.

Benefits of Distributed Packet Tracer

- Provides the possibility to execute from single point over NXAPI.
- Data Path traffic capture happens without the need to know internal architecture.
- Scheduled start and stop of a packet capture allows simultaneous start/stop on multiple devices.
- Decoding switch forwarding decision such as:
  - destination interface, VLAN
  - forward, drop
  - unicast, multi-destination (unknown-unicast, multicast, broadcast)

Supported Distributed Packet Tracer Configuration

Supported Hardware

DPT supports M3 and F3 series modules in Cisco NX-OS Release 8.2(1).

DPT supports only the below modules:

- N7K-M3xxx
- N77-M3xxx
- N7K-F3xxx
- N77-F3xxx

Supported Flow Filters

In Cisco NXOS Release 8.2(1) and in Cisco NX-OS Release 8.3(1), DPT implementation supports only the below filters:

- Classic Ethernet
  - L2 SRC/DST MAC
  - L3 SRC/DST IPv4, IPv6 address
  - IP protocol
  - VLAN

The above listed filters are supported on the FabricPath network (this does not include DFA), however filtering based on FTAG and FP TTL are not supported.

IP packet encapsulated in plain FabricPath header (this does not include DFA) is supported.

Only outer header filtering is supported. VXLAN/OTV/GRE inner IPv4/IPv6 filters are not supported. Filtering of MPLS encapsulated packets is not supported.

Configuration

DPT can be configured by:

- NXOS CLI
- NXAPI JSON
- NXAPI XML

You use the setup utility mainly for configuring the system initially, when no configuration is present. However, you can use the setup utility at any time for basic device configuration. The setup utility keeps the configured values when you skip steps in the script. For example, if you have already configured the mgmt0 interface, the setup utility does not change that configuration if you skip that step. However, if there is a default value for the step, the setup utility changes to the configuration using that default, not the configured value. Be sure to carefully check the configuration changes before you save the configuration.

Restrictions for Distributed Packet Tracer

Unsupported Hardware

In case of mixed chassis with supported and unsupported modules, DPT provides result only from the supported modules.

Timestamp

Timestamp presented in result CLI does not necessarily match the exact time when a packet arrives on the device. DPT checks hardware tables at specific intervals (default is 30 seconds). Therefore timestamp can be shifted by 30 seconds in comparison to actual time of packet arrival. Timestamp always references to local switch time.

Packet Count

Due to hardware limitations DPT can show only if the flow is present or not but it cannot count the number of packets transferred in the interval. If a specific flow is presented, packet count always shows one packet regardless of the number of packets sent through the switch.

Platform Limitations

DPT is mutually exclusive with ELAM feature. Any ELAM configuration will be overwritten by DPT and also manual ELAM execution can overwrite the applied DPT configuration. It is recommended not to use both features (DPT and ELAM) for troubleshooting at the same time because it provides incorrect results.

A few limitations can affect the accuracy of DPT due to the hardware architecture. When DPT does not capture traffic it does not mean that the packet did not arrive on the destination switch. There are chances that not all packets are received or forwarded.

The following scenarios/factors could occur due to the limitations impacting DPT:

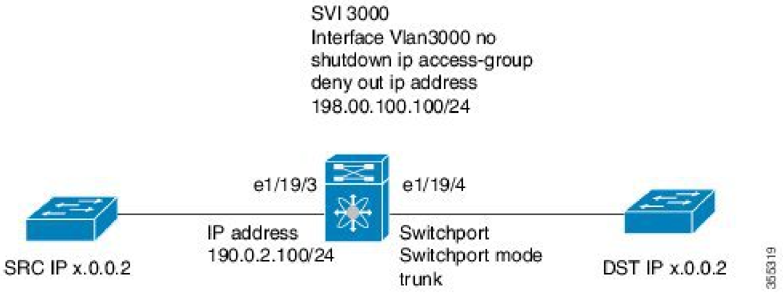
- Packet drops that occur inside packet buffers (ingress/egress/fabric) are not reflected in the final result. For example:
    - Packet drops in egress buffers (due to congestion) are shown as forwarded in the DPT.
    - Packet drops in ingress VOQ buffer (due to egress congestion) are shown as forwarded in the DPT.
- Decisions on egress forwarding ASIC are not reflected in the DPT. For example
    - Packet drops in egress PACL are shown in the DPT as forwarded. However, egress VACL is correctly shown as DROP since that decision happens in ingress ASIC.
    - Packets sent from CPU are not captured by the DPT. Only egress ASIC sees the outbound CPU packets.
- Current filtering capability supports only outer IP header filtering (packet encapsulated by OTV, VXLAN, GRE or DFA cannot be captured), any filter on MPLS encapsulated packets are not supported.
- The DPT flows that are created, their results and status are not persistent and is cleared upon SSO or upon the reload. All the created flows are cleared and need to be created and started again. Scheduled flow needs to be rescheduled.

How To Use The Distributed Packet Tracer

This section describes the standard workflow of Distributed Packet Tracer (DPT) usage.

To use DPT, **feature dpt** needs to be enabled in global configuration mode. Other commands are executed from the privilege EXEC mode.

Figure 3: Reference Topology for DPT



Configure and Start the DPT capture

Procedure

Step 1 Enable the DPT feature.

Example:

```
 Device(config)#feature dpt
Device(config)#
Device#
Device# show dpt ?
  flow     DPT flow
  results  Show results
  status   Status
Device#
```

Step 2 Create a flow; for example with a flow name, "first-flow" with a specific filter.

Example:

```
Device#dpt create flow first-flow src-ipv4 192.0.2.100 dst-ipv4 x.0.0.2

Flow first-flow created and in initialized status
Device#
Device# show dpt flow first-flow
----------------------------------------
ID: first-flow
Status: initialized
Definition:
  network-type classical-ethernet src-ipv4 192.0.2.100/32 dst-ipv4 x.0.0.2/32
 ---- System Admin Account Setup ----
```

Maximum of 10 flow definitions can be created. Capture is performed only on the ingress side.

After the creation flow status is in initialized status. This means that the flow is created in the supervisor database; however it is not installed in hardware. Multiple flows can be created.

It is recommended to use specific filters as much as possible; for example, use VLAN to capture traffic between layer 2 interfaces or in the fabric path network.

Step 3 Apply the newly created flow to the hardware.

Example:

```
Device#dpt apply flow first-flow

Flow first-flow applied and in configured status
Device# show dpt  flow first-flow
----------------------------------------
ID: first-flow
Status: configured
Definition:
  network-type classical-ethernet src-ipv4 192.168.208.109/32 dst-ipv4 50.0.0.2/3

Device# show dpt status flow first-flow
------------------------------------------------------------------------------------------
Flow                 Statistics Lookup-result  Status       Start-time           End-time
              Interval Detail
------------------------------------------------------------------------------------------
first-flow         n/a        n/a              configured
```

In the above example, flow has been installed in the hardware ASIC but result collection has not started. The state is similar to the ELAM when the trigger has been configured.

You can apply only one flow at a time in the hardware. You must release the old flow before applying a new flow.

Step 4 Start the flow capture.

Example:

```
Device#dpt start flow first-flow interval 10

Flow first-flow started and in armed status
Device# show dpt flow first-flow
----------------------------------------
ID: first-flow
Status: armed
Definition:
  network-type classical-ethernet src-ipv4 192.168.208.109/32 dst-ipv4 50.0.0.2/32

Device#
Device# show dpt status flow all
------------------------------------------------------------------------------------------
Flow                 Statistics Lookup-result  Status       Start-time           End-time
              Interval Detail
------------------------------------------------------------------------------------------
first-flow           n/a        n/a            armed        2017-09-05 06:06:19
            2017-09-05 10:06:19   10
Device#
```

DPT collects the results once the flow is started. Flow start and stop time can be specified in absolute calendar values or delay seconds from the current time.

In above example, the results collection happens in 10 second interval. The default results collection interval is 30 seconds, if not specified in the command. The capture time is limited to 4 hours by default from the start time, if not specified in the command. You must specify the start and end time if you need to run the capture for a longer time.

```
Device#dpt start flow first-flow start-time seconds 30 end-time 23:00:00 10
September 2017
Device#
Flow first-flow scheduled with start time

Device# show dpt flow first-flow
----------------------------------------
ID: first-flow
Status: armed
Definition: network-type classical-ethernet src-ipv4 192.168.208.109/32 dst-ipv4 50.0.0.2/32

Device# show dpt status flow first-flow

-----------------------------------------------------------------------------------------------
Flow                 Statistics Lookup-result  Status       Start-time            End-time
              Interval Detail
-----------------------------------------------------------------------------------------------
first-flow          n/a       n/a           armed       2017-09-05 06:12:15   2017-09-05
 10:12:15   10
```

You can apply only one flow at a time in hardware. You must stop and release the already captured flow before applying a new flow.

Show Capture Results

Procedure

	Command or Action	Purpose

Step 1	Verify if the flow is started/armed.
	Example:
	`Device# show dpt status flow all`
	`----------------------------------------`
	`Flow                  Statistics`
	`Lookup-result  Status       Start-time`
	`          End-time             Interval`
	` Detail`
	`----------------------------------------`
	`first-flow           n/a        n/a`
	`        armed       2017-09-05 11:44:30`
	`  2017-09-05 15:44:30   10`	

Step 2	Verify the capture results. **Example:** `Device# show dpt result flow first-flow`	

	Command or Action	Purpose

```
Flow ID: first-flow   Start-time
[2017-09-05 11:52:20]  End-time
[2017-09-05 15:52:20]  Interval [10]

Idx  |Result|Drop  |      Timestamp
   |Input
                     |Output

     |      |reason|
   |interface            |Vlan  |BD
|VNI   |Rate  |Count |interface
    |Vlan  |BD    |VNI   |Rate  |Count
-------------------------------------------
2      fwd    n/a   2017-09-05 11:53:00
    Ethernet1/19/4        3000    n/a
  n/a    n/a   1      Ethernet1/19/3
     0       n/a    n/a    n/a   1
1      fwd    n/a   2017-09-05 11:52:50
    Ethernet1/19/4        3000    n/a
  n/a    n/a   1      Ethernet1/19/3
     0       n/a    n/a    n/a   1
0      fwd    n/a   2017-09-05 11:52:40
    Ethernet1/19/4        3000    n/a
  n/a    n/a   1      Ethernet1/19/3
     0       n/a    n/a    n/a   1
```

Results are collected in 10 seconds interval; maximum 180 results are stored per flow.

When DPT cannot decode a result it will show as "n/a".

These results support XML/JSON format. DPT also supports NXAPI for remote execution from NMS.

Step 3 Verify the detailed results.

Example:

```
Device# show dpt results flow first-flow
 detail


-----------------------------------------------
Result details for flow ID: first-flow
-----------------------------------------------
Index                        1
Timestamp                    2017-09-21
22:21:55
Source Interface            Ethernet1/30
Source MAC address
6c20.56e8.4f3c
Source IP address            x.1.1.2

Destination Interface       Ethernet2/11
Destination MAC address
0026.51c7.fcc1
Destination IP address       x.1.1.1
```

	Command or Action	Purpose

```
IP Protocol                  1
Source L4 port               0
Destination L4 port          0
Source Vlan ID               133
Destination Vlan ID          133
Source Bridge Domain         n/a
Destination Bridge Domain    n/a
Source VNI                   n/a
Destination VNI              n/a

---------------------------------------------------
Index                        0
Timestamp                    2017-09-21
22:21:25
Source Interface            Ethernet1/30
Source MAC address
6c20.56e8.4f3c
Source IP address            x.1.1.2

Destination Interface       Ethernet2/11
Destination MAC address
0026.51c7.fcc1
Destination IP address       x.1.1.1

IP Protocol                  1
Source L4 port               0
Destination L4 port          0
Source Vlan ID               133
Destination Vlan ID          133
Source Bridge Domain         n/a
Destination Bridge Domain    n/a
Source VNI                   n/a
Destination VNI              n/a
```

Stop and Release the Capture

Procedure

	Command or Action	Purpose
Step 1	Stop the flow.	

Example:

```
Device# dpt stop flow first-flow
Flow first-flow stopped and in configured
 status
Device# show dpt status flow all
────────────────────────────────────────────────
Flow                 Statistics
Lookup-result  Status       Start-time
          End-time             Interval
 Detail
────────────────────────────────────────────────
first-flow           n/a        n/a
        configured
```

	Command or Action	Purpose
	Results are not cleared when the flow capture is stopped.	

Step 2	Release the flow.
	Example:
	`Device# dpt release flow first-flow`
	`Flow first-flow released and in initialized status`
	`Device#`
	Results are not cleared when the flow capture is released.	

Step 3	Delete the flow.
	Example:
	`Device# dpt delete flow first-flow`
	`Flow first-flow deleted`
	`Device#`
	Deleting the flow will delete all the results.	

Configuration Example for the Distributed Packet Tracer

Example: Multi-destination Result

The following example shows the shows the multi-destination case (unknown-unicast, multicast, and broadcast).

```
Device# show dpt result flow first-flow
------------------------------------------------------------------------------------------
Flow ID: first-flow   Start-time [2017-09-05 11:52:20]  End-time [2017-09-05 15:52:20]
Interval [10]
------------------------------------------------------------------------------------------
Idx  |Result|Drop  |      Timestamp       |Input
         |Output
     |      |reason|                      |interface           |Vlan  |BD    |VNI   |Rate
  |Count |interface              |Vlan  |BD    |VNI   |Rate  |Count
------------------------------------------------------------------------------------------
1     fwd    n/a    2017-08-24 14:04:25    Ethernet1/19/3        0       n/a    n/a    n/a
   1      multi-dest LTL_0xc019   3000    n/a    n/a    n/a   1
```

In this example, the output interfaces are not listed as the traffic is forwarded to multiple destination ports; only the internal port index (LTL) is specified.

The following example provides a list of specific interfaces:

```
Device# show system internal pixm info ltl 0xc019

LTL      res_id          ltl_flag        cb_flag         MI[0]
```

```
0xc019   0x00000000      0x00000000      0x00000000      0x0fff

Member info
------------------
IFIDX           LTL
---------------------------------
Eth101/1/8         0x252c
Eth101/1/14        0x2532
Eth101/1/2         0x2526
Eth101/1/4         0x2528
...
Po101              0x0e00
Eth102/1/2         0x2586
Eth102/1/7         0x258b
Eth1/19/4          0x0bde
Eth102/1/8         0x258c
Eth102/1/9         0x258d
```

Example: Drop Result

The following example shows the drop result when the traffic is dropped by the egress VACL on SVI 3000.

```
Device# show dpt result flow first-flow
------------------------------------------------------------------------------
Flow ID: first-flow   Start-time [2017-09-05 11:52:20]  End-time [2017-09-05 15:52:20]
Interval [10]

Idx  |Result|Drop  |      Timestamp       |Input
         |Output
     |      |reason|                      |interface            |Vlan  |BD    |VNI   |Rate
  |Count |interface              |Vlan  |BD    |VNI   |Rate  |Count

1     drop   n/a   2017-08-24 14:04:25    Ethernet1/19/3         0       n/a    n/a    n/a
  1       Drop LTL:0xcad          3000    n/a    n/a    n/a   1
```

Drop reason decode is not supported in Cisco NX-OS Release 8.2(1). Perform a manual traffic forwarding result analysis to determine the exact drop reason with the assistance of Cisco TAC.

Example: Unknown Result

In corner cases DPT might not be able to identify if packet has been forwarded or dropped. In such a case the result status has "n/a" field and the output interface has the destination LTL index. For these cases, additional manual traffic analysis is required with the assistance of Cisco TAC.

```
Device# show dpt result flow first-flow

------------------------------------------------------------------------------
Flow ID: first-flow   Start-time [2017-09-05 11:52:20]  End-time [2017-09-05 15:52:20]
Interval [10]

Idx  |Result|Drop  |      Timestamp       |Input
         |Output
     |      |reason|                      |interface            |Vlan  |BD    |VNI   |Rate
  |Count |interface              |Vlan  |BD    |VNI   |Rate  |Count

1     n/a   n/a   2017-08-24 14:04:25    Ethernet1/19/3         0       n/a    n/a    n/a
 1        LTL_0xccc           3000    n/a    n/a    n/a   1
```

Drop reason decode is not supported in Cisco NX-OS Release 8.2(1). Perform a manual traffic forwarding result analysis to determine the exact drop reason with the assistance of Cisco TAC.

CHAPTER 6

Network Plug and Play

This chapter provides information about the Network Plug and Play (PnP) feature in the Cisco Nexus 7000 Series Switches, and contains the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Feature History for Network Plug and Play

This table lists the release history for this feature.

Table 6: Feature History for Network Plug and Play

Feature Name	Releases	Feature Information	
Network Plug and Play	8.2(1)	This feature was introduced. Network Plug and Play (PnP) is a software application that runs on a Cisco Nexus 7000 Series Switch. The PnP feature provides a simple, secure, unified, and integrated offering to make a new branch or campus rollouts much easier, or for provisioning updates to an existing or a new network. This feature provides a unified approach to provision networks comprising multiple devices with a near-zero-touch deployment experience.	

Information About Network Plug and Play

Network Plug and Play (PnP) is a software application that runs on a Cisco Nexus 7000 Series Switch. The PnP feature provides a simple, secure, unified, and integrated offering to make a new branch or campus rollouts much easier, or for provisioning updates to an existing or a new network. This feature provides a unified approach to provision networks comprising multiple devices with a near-zero-touch deployment experience.

Simplified deployment reduces the cost and complexity and increases the speed and security of the deployments. The PnP feature helps simplify the deployment of any Cisco device by automating the following deployment-related operational tasks:

- Establishing initial network connectivity for a device.
- Delivering device configuration to the controller.
- Delivering software and firmware images to the controller.
- Delivering licenses to the controller.
- Delivering deployment script files to the controller.
- Provisioning local credentials of a switch.
- Notifying other management systems about deployment-related events.

The PnP is a client-server based model. The client (agent) runs on a Cisco Nexus 7000 Series Switch and the server (controller) runs on the Cisco DNA Controller.

PnP uses a secure connection to communicate between the agent and the controller. This communication is encrypted.

The PnP agent converge solutions that exist in a network into a unified agent and adds additional functionality to enhance the current solutions. The main objectives of the PnP agent are:

- Provide consistent Day 0 deployment solution for all the deployment scenarios.
- Add new or required features to improve existing solutions.
- Provide Day 2 management framework mainly in the context of configuration and image upgrades.

Features Provided by the Network Plug and Play (PnP) Agent

Some of the features that the PnP agent provides are:

- Day 0 bootstrapping. This includes the configuration, image, licenses, and other files.
- Day 2 management. This includes the configuration and image upgrades and ongoing monitoring of Simple Network Management Protocol (SNMP) and syslog messages.
- Open communication protocol. This enables customers and partners to write applications.
- XML-based payload over HTTP.
- Security. This includes authentication and encrypted communication channel between the management app and the agent.
- Deployment and management of devices behind firewall and Network Address Translation (NAT).
- Support for one-to-one and one-to-many communication.
- Support for policy-based deployment (product ID or location of the device).
- Deployment based on unique ID (Unique Device Identifier [UDI] or MAC).
- Support for various deployment scenarios and use cases.
- Zero-touch deployment is performed whenever possible. Low-touch deployment is performed based on the need.

When a device is powered on for the first time, the PnP discovery process, which is embedded in the device, gets enabled in the absence of a startup configuration file and attempts to discover the address of the PnP controller or server. The PnP agent uses methods such as DHCP, Domain Name System (DNS), and others to acquire the desired IP address of the PnP server.

When the PnP agent successfully acquires the IP address, it initiates a long-term, bidirectional Layer 3 connection with the server and waits for a message from the server. The PnP server application sends messages to the corresponding agent, requesting for information about the devices and the services to be performed on the device.

The agent running on the Cisco Nexus 7000 Series switch then configures the IP address on receiving the DHCP acknowledgment and establishes a secure channel with the controller to provision the configurations. The switch then upgrades the image and applies the configurations.

Discovery Methods

A PnP agent discovers the PnP controller or server using one of the following methods:

- DHCP-based discovery
- DNS-based discovery
- PnP connect

After the discovery, the PnP agent writes the discovered information into a file, which is then used to handshake with the PnP server (DNA controller/APIC-EM).

The following tasks are carried out by the agent in the PnP discovery phase:

- Brings up all the interfaces.
- Sends a DHCP request in parallel for all the interfaces.
- On receiving a DHCP reply, configures the IP address and mask, default route, DNS server, domain name, and writes the PnP server IP in a lease-parsing file. Note that there is no DHCP client in Cisco Nexus Switches and static configuration is required.
- Brings down all the interfaces.

DHCP-Based Discovery

When the switch is powered on and if there is no startup configuration, the PnP starts with DHCP discovery. DHCP discovery obtains the PnP server connectivity details.

The PnP agent configures the following:

- IP address
- Netmask
- Default gateway
- DNS server
- Domain name

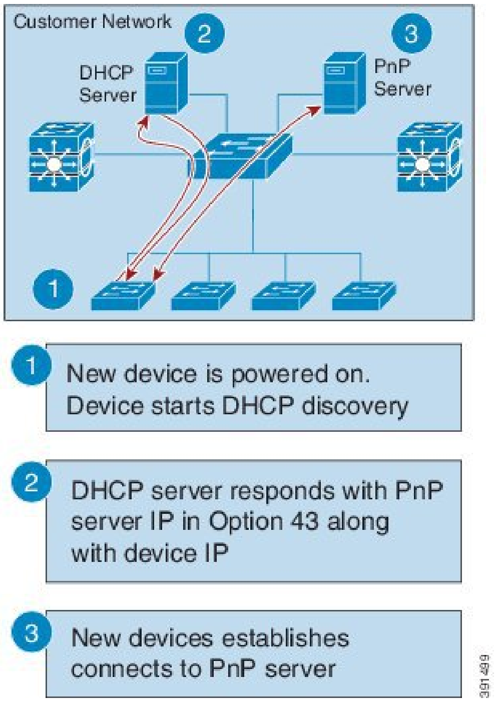
If the agent configuration fails, you should manually intervene and configure the switch.

DHCP discovery has the following flow:

- Power on the switch.
- Switch will boot up, the PnP process will be started, as there is no configuration present.
- Start DHCP discovery.
- DHCP Server replies with the PnP agent and the PnP server configuration.
- PnP agent handshakes with the PnP server.
- Download the image, install, and reload.
- Download and apply the configuration from the controller.
- Reload the switch.

A device with no startup configuration in the NV-RAM triggers the PnP agent to initiate a DHCP discovery process, which acquires the IP configuration from the DHCP server required for the device. The DHCP server can be configured to insert additional information using vendor-specific Option 43. Upon receiving Option 60 from the device with the string (cisco pnp), to pass on the IP address or hostname of the PnP server to the requesting device. When the DHCP response is received by the device, the PnP agent extracts the Option 43 from the response to get the IP address or the hostname of the PnP server. The PnP agent then uses this IP address or hostname to communicate with the PnP server.

Figure 4: DHCP Discovery Process for PnP Server



1 New device is powered on. Device starts DHCP discovery

2 DHCP server responds with PnP server IP in Option 43 along with device IP

3 New devices establishes connects to PnP server

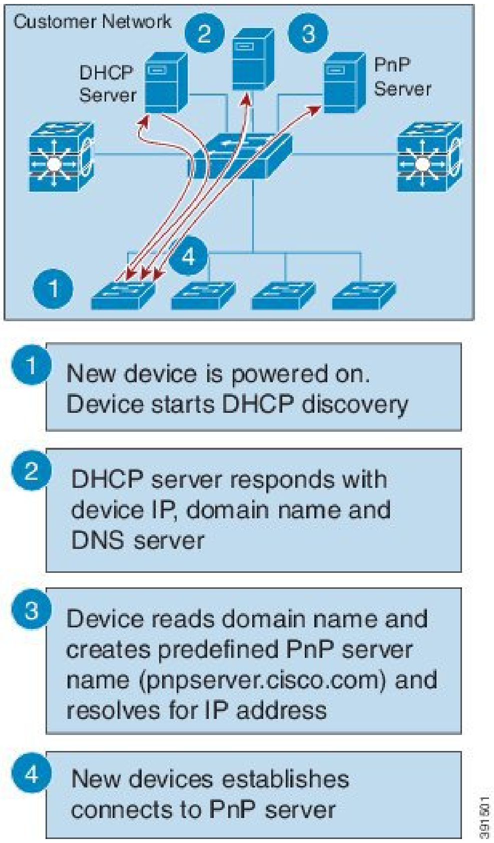
DNS-Based Discovery

When the DHCP discovery fails to get the PnP server, the agent falls back to DNS-based discovery. To start the DNS-based discovery, the following information is required from DHCP:

- IP address and netmask
- Default gateway
- DNS server IP
- Domain name

The agent obtains the domain name of the customer network from the DHCP response and constructs the fully qualified domain name (FQDN). The following FQDN is constructed by the PnP agent using a preset deployment server name and the domain name information for the DHCP response. The agent then looks up the local name server and tries to resolve the IP address for the above FQDN.

Figure 5: DNS Lookup for pnpserver.[domainname].com


Customer Network
2
3
DHCP Server
PnP Server
4
1
1 New device is powered on. Device starts DHCP discovery
2 DHCP server responds with device IP, domain name and DNS server
3 Device reads domain name and creates predefined PnP server name (pnpserver.cisco.com) and resolves for IP address
4 New devices establishes connects to PnP server
391501

Note The device reads domain name and creates predefined PnP server name as pnpserver.[domain name].com, for example; pnpserver.cisco.com.

Plug and Play Connect

When the DHCP and the DNS discovery fail, the PnP agent discovers and communicates with Cisco Cloud-based deployment service for initial deployment. The PnP agent directly opens an HTTPS channel using the Python library, which internally invokes OpenSSL to talk with cloud for configuration.

Cisco Power On Auto Provisioning

Cisco Power On Auto Provisioning (PoAP) communicates with the DHCP and TFTP servers to download the image and configurations. With the introduction of the PnP feature, PnP and PoAP coexist together in a Cisco Nexus 7000 switch. PoAP and PnP interworking has the following processes:

- PoAP starts first no configuration is present in the system.
- PnP starts later if PoAP does not get provisioned.
- PoAP and PnP discover the controller alternatively.
- The controller discovery process continues until a controller or until the admin aborts auto provision.
- The process (POAP or PnP) that finds the controller continues provisioning and the other process that does not find the controller is notified and eventually terminated.

Services and Capabilities of the Network Plug and Play Agent

The PnP agent performs the following tasks:

- Backoff
- Capability
- CLI execution
- Configuration upgrade
- Device information
- Certificate install
- Image install
- Redirection

Note The PnP controller or server provides an optional checksum tag to be used in the image installation and configuration upgrade service requests by the PnP agent. When the checksum is provided in a request, the image install process compares the checksum against the current running image checksum.

If the checksums are same, the image being installed or upgraded is the same as the current image running on the device. The image install process will not perform any other operation in this scenario.

If the checksums are not the same, the new image will be copied to the local file system, and the checksum will be calculated again and compared with the checksum provided in the request. If they are the same, the image install process continues to install the new image or upgrade the device to the new image. If the checksums are not the same, the process exits with an error.

Backoff

A Cisco NX-OS device that supports PnP protocol, which uses HTTP transport, requires the PnP agent to send the work request to the PnP server continuously. If the PnP server does not have any scheduled or outstanding PnP service for the PnP agent to execute, the continuous no-operation work requests exhaust both the network bandwidth and the device resources. This PnP backoff service allows the PnP server to inform the PnP agent to rest for the specified time and call back later.

Capability

Capability service request is sent by the PnP server to the PnP agent on a device to query the supported services by the agent. The server then sends an inventory service request to query the device's inventory information; and then sends an image installation request to download an image and install it. After getting the response from the agent, the list of supported PnP services and features are enlisted and returned back to the Server.

CLI Execution

Cisco NX-OS supports two modes of command execution, privileged EXEC mode and global configuration mode. Most of the EXEC commands are one-time commands, such as **show** commands, which show the current configuration status, and clear commands, which clear counters or interfaces. The EXEC commands are not saved when a device reboots. Configuration mode commands allow user to make changes to the running configuration. If you save the configuration, these commands are saved when a device reboots.

Configuration Upgrade

Two types of configuration upgrades takes place in a Cisco device—copying new configuration files to the startup configuration and copying new configuration files to the running configuration.

Copying new configuration files to the startup configuration—A new configuration file is copied from the file server to the device using the **copy** command, and the file check task is performed to check the validity of the file. If the file is valid, the file is copied to the startup configuration. The previous configuration file is backed up if enough disk space is available. The new configuration comes into effect when the device reloads again.

Copying new configuration files to the running configuration—A new configuration file is copied from the file server to the device using the **copy** command or **configure replace** command. Replace and rollback of configuration files may leave the system in an unstable state if rollback is performed inefficiently. Therefore, configuration upgrade by copying the files is preferred.

Device Information

The PnP agent provides the capability to extract device inventory and other important information to the PnP server on request. The following device-profile request types are supported:

- all—Returns complete inventory information, which includes unique device identifier (UDI), image, hardware, and file system inventory data.
- filesystem—Returns file system inventory information, which includes file system name and type, local size (in bytes), free size (in bytes), read flag, and write flag.
- hardware—Returns hardware inventory information, which includes hostname, vendor string, platform name, processor type, hardware revision, main memory size, I/O memory size, board ID, board rework ID, processor revision, mid-plane revision, and location.
- image—Returns image inventory information, which includes version string, image name, boot variable, return to ROMMON reason, bootloader variable, configuration register, configuration register on next boot, and configuration variables.
- UDI—Returns the device UDI.

Certificate Install

Certificate install is a security service through which a PnP server requests the PnP agent on a device for trust pool or trust point certificate installation or uninstallation. This service also specifies the agent about the primary and backup servers for reconnection. The following prerequisites are required for a successful certificate installation:

- The server from which the certificate or trust pool bundle needs to be downloaded should be reachable.
- There should not be any permission issues to download the certificate or the bundle.
- The PKI API should be available and accessible for the PnP agent so that the agent could call to download and install the certificate or the bundle.
- There is enough memory on the device to save the downloaded certificate or bundle.

Image Install

The image install service enables a PnP-enabled device to perform image upgrade on receiving a request from the PnP server.

An Image Install request can be made for the following types of devices:

- Standalone devices
- High-availability devices
- Stackable devices
- Cisco Nexus 7000 Series devices

Standalone Devices

When the PnP agent on a standalone device receives a request from the PnP server, the agent parses the XML payload and identifies the request as an Image Upgrade request. The agent then creates an ImageInstall process, which identifies the request as a standalone image install request.

High-Availability Devices

When the PnP agent is installed on a high-availability device, and the ImageInstall service gets the data structure, the agent determines if the request is for a high-availability device. The active route processor (RP) that is running the PnP agent performs all the tasks required to install the image on both the active and standby devices.

Redirection

The Redirection service is used to redirect a device to another controller.

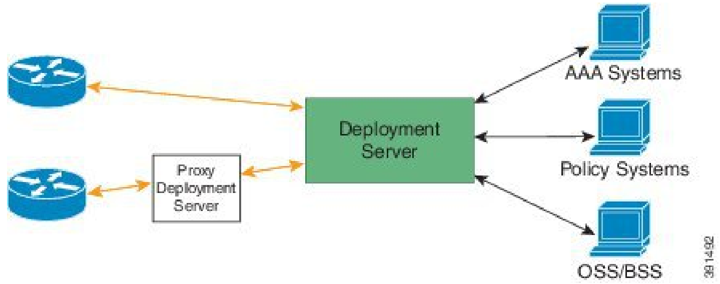
PnP Agent

The PnP agent is an embedded software component that is present in all Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent first tries to discover a PnP server, with which it can communicate. After a server is found and connection established, the agent performs deployment-related activities such as configuration, image, license, and file updates by communicating with the server. It also notifies the server of all interesting deployment-related events such as out-of-band configuration changes and new device connections on an interface.

PnP Server

The PnP server is a central server that encodes the logic of managing and distributing deployment information (images, configurations, files, and licenses) for the devices being deployed. The server communicates with the agent on the device that supports the simplified deployment process using a specific deployment protocol.

Figure 6: Simplified Deployment Server



The PnP server also communicates with proxy servers such as deployment applications on smart phones and PCs, or other PnP agents acting as Neighbor Assisted Provisioning Protocol (NAPP) servers, and other types of proxy deployment servers such as VPN gateways.

The PnP server can redirect the PnP agent to another deployment server. A common example of redirection is a PnP server redirecting a device to communicate with it directly after sending the bootstrap configuration through a NAPP server. A PnP server can be hosted by an enterprise. This solution allows for a cloud-based deployment service provided by Cisco. In this case, a device discovers and communicates with Cisco cloud-based deployment service for initial deployment. After that, it can be redirected to the customer's deployment server.

In addition to communicating with the devices, the server interfaces with a variety of external systems such as authentication, authorizing, and accounting (AAA) systems, provisioning systems, and other management applications.

PnP Agent Deployment

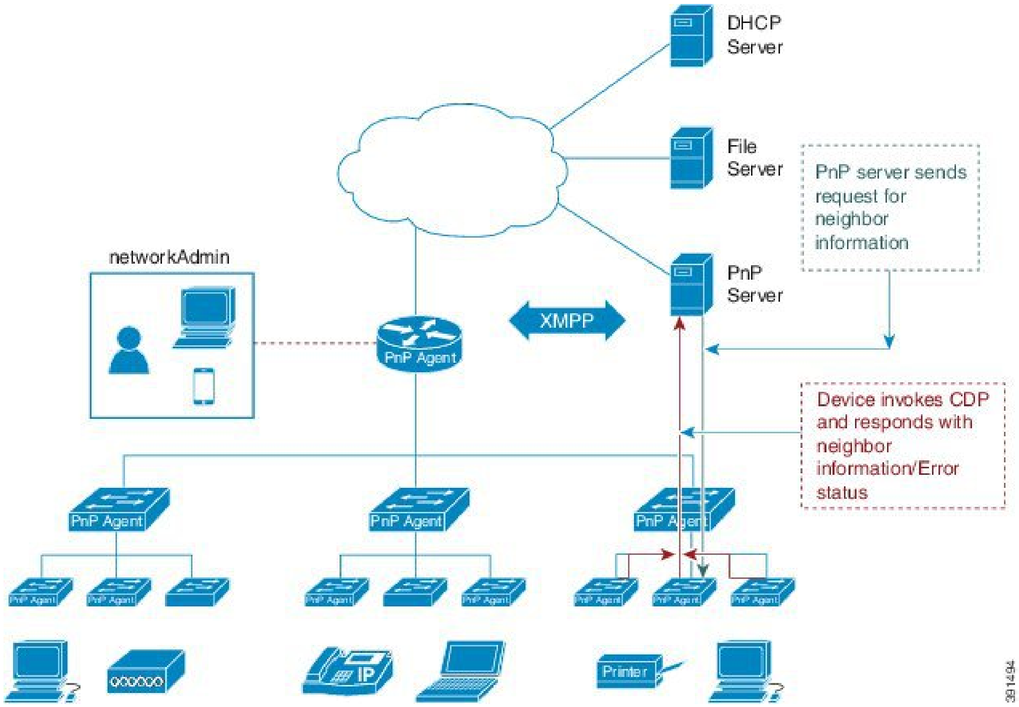
The following steps indicate the PnP agent deployment procedure on Cisco devices:

1. A Cisco device with a PnP agent contacts the PnP server, requesting for a task, that is, the PnP agent sends UDI along with a request for work.

2. If the PnP server has a task for the device, for example, image installation, configuration, upgrade, and so on, it sends a work request.
3. After the PnP agent receives the work request, it executes the task and sends back a reply to the PnP server about the task status, that is whether it is successful or if an error has occurred, and the corresponding information that is requested.

PnP Agent Network Topology

Figure 7: Network Topology of Cisco PnP Agent Deployment


DHCP Server
File Server
PnP server sends request for neighbor information
PnP Server
networkAdmin
XMPP
PnP Agent
Device invokes CDP and responds with neighbor information/Error status
PnP Agent
PnP Agent
PnP Agent
PnP Agent
IP
Printer
391494

PnP Agent Initialization

The PnP agent is enabled by default, but can be initiated on a device when the startup configuration is not available.

Absence of Startup Configuration

New Cisco devices are shipped to customers with no startup configuration file in the NVRAM of the devices. When a new device is connected to a network and powered on, the absence of a startup configuration file on the device automatically triggers the PnP agent to discover the PnP server IP address.

CLI Configuration for the PnP Agent

PnP supports devices that are using VLAN 1 by default. To use a VLAN other than 1, adjacent upstream devices must configure the **pnp startup-vlan** *vlan-id* command on the upstream device.

This configuration on the upstream switch directs the VLAN that needs to be configured by the downstream switch for PnP provisioning. The VLAN value is exchanged with the downstream switch using Cisco Discovery Protocol (CDP) type, length, values (TLVs). All the inband ports of the downstream switch are configured as a trunk on receiving the **pnp startup vlan** from CDP TLV for Day 0 provisioning.

Guidelines for the PnP Deployment

- The PnP deployment method depends on the discovery process required for finding the PnP controller or server.
- The discovery mechanism should be deployed, either as a DHCP server discovery process or a Domain Name Server (DNS) discovery process, before launching PnP.
- The DHCP server or the DNS server should be configured before deploying PnP.
- The PnP server should communicate with the PnP agent.
- PnP connect does not require a DHCP or DNS configuration.
- PnP runs both the in-band and the management interfaces.
- IPv6 support for PnP is not available for Cisco Nexus 7000 Series devices.
- The kickstart and system images must be bundled into a tar file to update in APIC-EM.
- The bootflash should have enough space to download the image and configurations from APIC-EM.

Configuring the Upstream Switch to Broadcast PnP

Configure and Start the DPT capture

Procedure

Step 1 Enable the global configuration mode.

Example:

```
 switch#configure terminal
```

Step 2 Configure the upstream switch to broadcast PnP VLAN over the Cisco Discovery Protocol (CDP):

Example:

```
switch(config)# pnp startup-vlan vlan ID
```

Note To use a VLAN other than 1, adjacent upstream devices must configure the **pnp startup-vlan** *vlan-id* command on the upstream device. This configuration must be performed to push this command to the upcoming PnP device.

When you execute the **pnp startup-vlan** *vlan-id* command on an adjacent upstream device, the VLAN membership change does not happen on that device. However, all the active interfaces on the upcoming PnP device are changed to the specified VLAN.

Step 3 Exit global configuration mode and enter privileged EXEC mode:

Example:

```
switch(config)#end
```

Step 4 Verify the PnP status.

Example:

```
switch# show pnp status
```

Step 5 Display the PnP summary.

Example:

```
switch# show pnp summary
```

Step 6 Display the configured PnP profiles.

Example:

```
switch# show pnp profiles
```

Step 7 Troubleshoot PnP using these commands.

Example:

```
switch#show pnp internal info
switch#show pnp internal stats
switch#show logging log | grep -i pnp
switch#Show pnp internal trace
switch#show pnp internal msgs
switch# show tech-support pnp
```

Configuration Examples for Network Plug and Play

Example: Troubleshooting PnP

The following examples shows the PnP troubleshooting command outputs:

```
Switch# show pnp internal info

PnP Global Information
UDI:
VDC: switch
Platform: N77
Serial Number: FXS1820Q0MQ
Product ID: N77-C7706
 Software Version: 8.2(0)SK(1)
State:
PnP Phase: Init
FSM State: PNP_STATE_INIT
PnP Accelerated: No
Global Variable:
MTS Q FD: 11
```

```
Switch# show logging log | grep -i pnp

2017 Jan  3 13:01:42 switch %PNP-2-PNP_INFO: PnP Ignited
2017 Jan  3 13:01:59 switch %PNP-2-PNP_INFO: PnP Accelerated
2017 Jan  3 13:03:00 switch %PNP-2-PNP_INFO: PnP Starting DHCP Discovery
2017 Jan  3 13:03:01 switch %PNP-2-PNP_INFO: PnP Received Valid Offer, Saved.
2017 Jan  3 13:03:01 switch %PNP-2-PNP_INFO: PnP Received Best Offer, Saved.
2017 Jan  3 13:03:10 switch %PNP-2-PNP_INFO: Configuring IP Address from DHCP
```

```
Switch# show pnp internal stats

PnP Status
Invalid Argument : 0
No Memory : 0API Failed : 0
Net L2 Reg Failed : 0
Device Discovey Failed : 0
Pump Failed : 0
Create Event Faild : 0
Tx Failed : 0
Timer Faild : 0
```

CHAPTER 7

Using PowerOn Auto Provisioning

This chapter describes how to deploy and use PowerOn Auto Provisioning (POAP) for the Cisco Nexus 7000 Series device.

This chapter contains the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Guidelines and Limitations for POAP

POAP configuration guidelines and limitations are as follows:

- The Cisco Nexus switch software image must support POAP for this feature to function.
- POAP does not support provisioning of the switch after it has been configured and is operational. Only auto-provisioning of a switch with no startup configuration is supported.
- If you use POAP to bootstrap a Cisco Nexus device that is a part of a vPC (virtual port channel) pair using static port channels on the vPC links, the Cisco Nexus device activates all of its links when POAP starts up. The dually connected device at the end of the vPC links might start sending some or all of its traffic to the port-channel member links that are connected to the Cisco Nexus device, which causes traffic to get lost.

To work around this issue, you can configure the Link Aggregation Control Protocol (LACP) on the vPC links so that the links do not incorrectly start forwarding traffic to the Cisco Nexus device that is being bootstrapped using POAP.

- If you use POAP to bootstrap a Cisco Nexus device that is connected downstream to a Cisco Nexus 7000 Series device through a LACP port channel, the Cisco Nexus 7000 Series device defaults to suspend its member port if it cannot bundle it as a part of a port channel. To work around this issue, configure the Cisco Nexus 7000 Series device to not suspend its member ports using the **no lacp suspend-individual** command from interface configuration mode.
- To support POAP to be more secure, ensure that DHCP snooping is enabled; and set the firewall rules to block unintended or malicious DHCP servers.
- When you reload a system with Cisco NX-OS Release 8.3(1) and when you abort POAP using “Ctrl+C” after a write-erase reload, POAP will crash.
- POAP with v6 is supported only with the IPv6 link-local address as the next-hop. This is a day-1 limitation.
- Important POAP updates are logged in the syslog and are available from the serial console.
- Critical POAP errors are logged to the bootflash. The filename format is *date-time* _poap_*PID*_[init,1,2].log, where *date-time* is in the YYYYMMDD_hhmmss format and *PID* is the process ID.
- Script logs are saved in the bootflash directory. The filename format is *date-time*_poap_*PID*_script.log, where *date-time* is in the YYYYMMDD_hhmmss format and *PID* is the process ID.

  You can configure the format of the script log file. Script file log formats are specified in the script. The template of the script log file has a default format; however, you can choose a different format for the script execution log file.
- The POAP feature does not require a license and is enabled by default. However for the POAP feature to function, appropriate licenses must be installed on the devices in the network before the deployment of the network.

Note To allow the POAP feature to function temporarily without the installation of the appropriate licenses, you can specify the **license grace-period** command in the configuration file.

This workaround allows you to install the appropriate licenses at a later time.

Information About PowerOn Auto Provisioning

PowerOn Auto Provisioning (POAP) automates the process of upgrading software images and installing configuration files on Cisco Nexus switches that are being deployed in the network for the first time.

When a Cisco Nexus Series switch with the POAP feature boots and does not find the startup configuration, the switch enters POAP mode, locates a DHCP server, and bootstraps itself with its interface IP address, gateway, and DNS server IP addresses. The switch also obtains the IP address of a TFTP server or the URL of an HTTP server and downloads a configuration script that enables the switch to download and install the appropriate software image and configuration file.

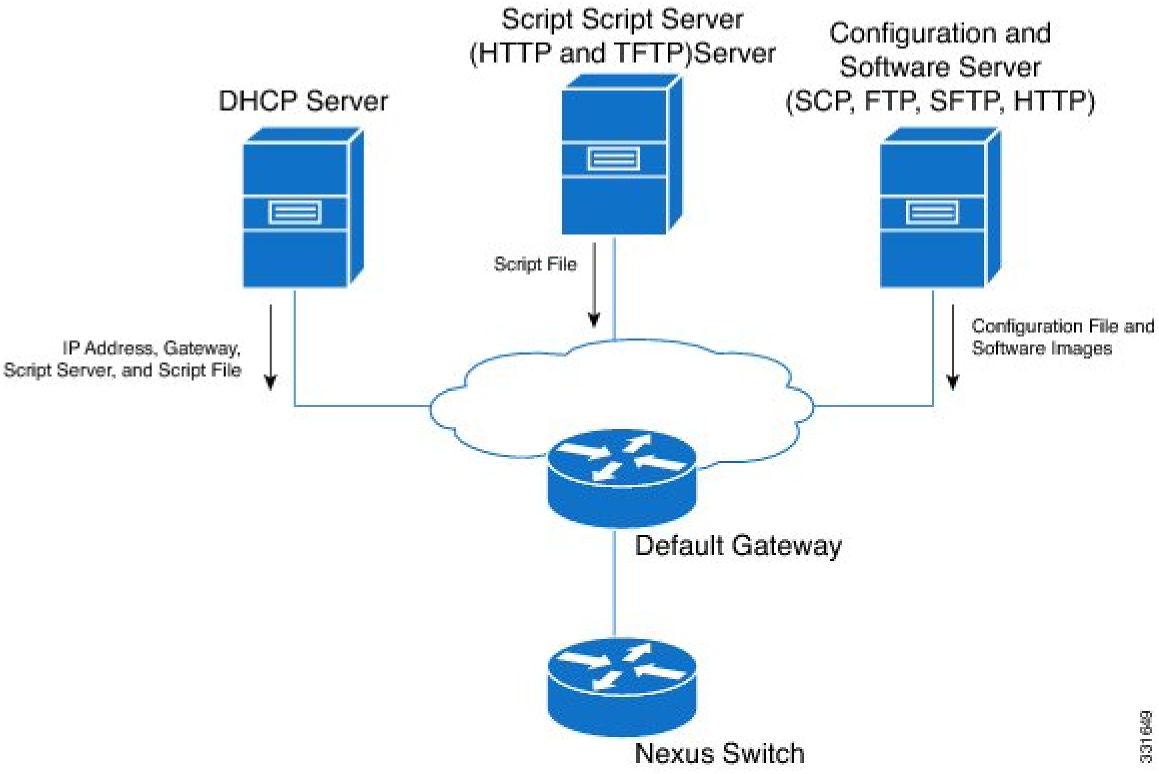
Note The DHCP information is used only during the POAP process.

Network Requirements for POAP

If a USB (Universal Serial Device) device that contains the required installation files is not available, POAP requires the following network infrastructure:

- A DHCP server to bootstrap the interface IP address, gateway address, and DNS (Domain Name System) server.
- A TFTP server that contains the configuration script used to automate the software image installation and configuration process.
- One or more servers that contains the desired software images and configuration files.

Figure 8: POAP Network Infrastructure



POAP Configuration Script

The reference script supplied by Cisco supports the following functionality:

- Retrieves the switch-specific identifier, for example, the serial number.

- Downloads the software image (system and kickstart images) if the files do not already exist on the switch. The software image is installed on the switch and is used at the next reboot.
- Schedules the downloaded configuration to be applied at the next switch reboot.
- Stores the configuration as the startup configuration.

Cisco has sample configuration scripts that were developed using the Python programming language and Tool Command Language (Tcl). You can customize one of these scripts to meet the requirements of your network environment.

For Cisco Nexus 7000 Series devices, the Python programming language uses two APIs that can execute CLI commands. These APIs are described in the following table. The arguments for these APIs are strings of the CLI commands.

API	Description
cli()	Returns the raw output of CLI commands, including the control/special characters.
clid()	For CLI commands that support XML, this API puts the command output in a Python dictionary. This API can be useful to help search the output of **show** commands.

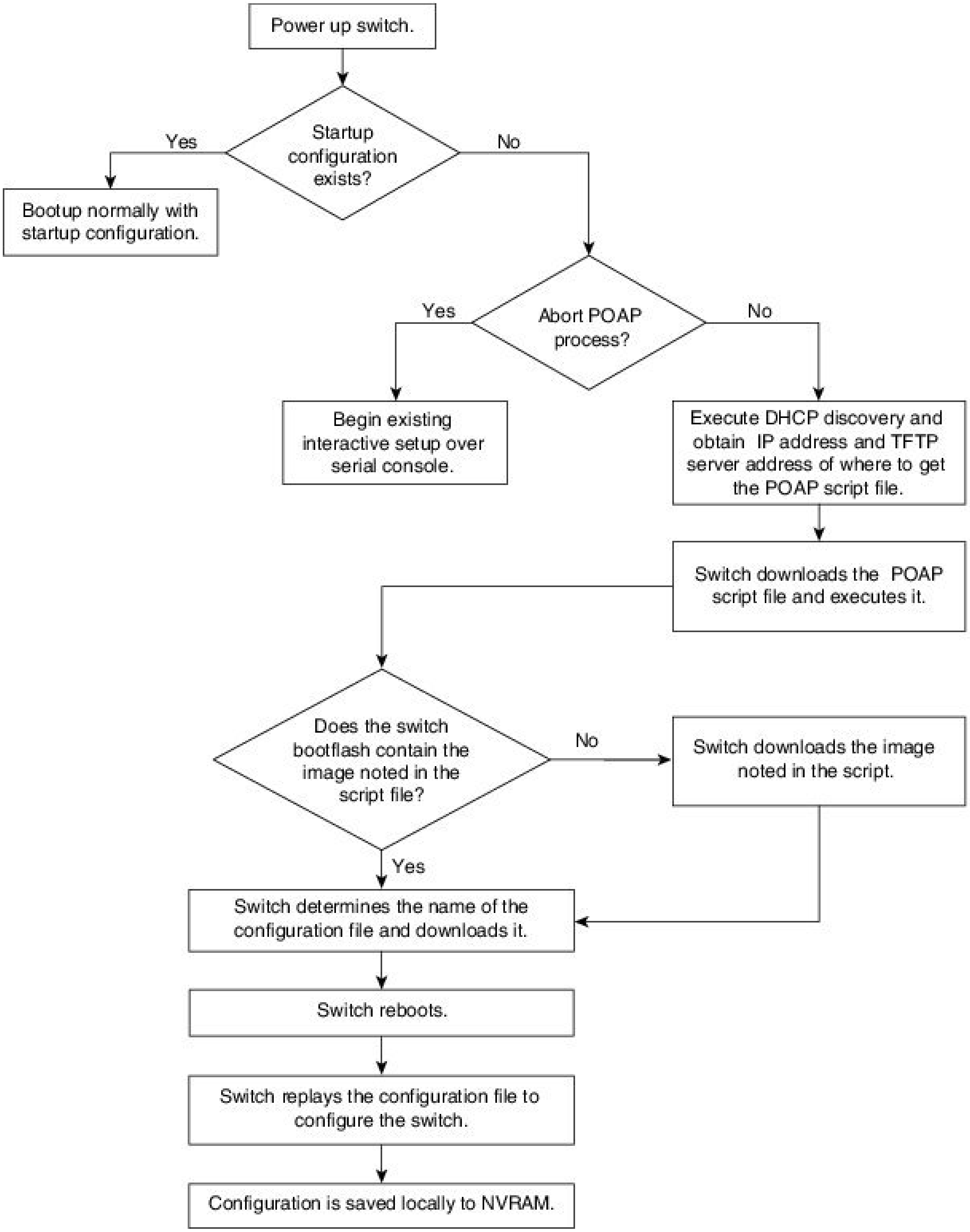
POAP Process

The POAP process has the following phases:

1. Power up
2. USB discovery
3. DHCP discovery
4. Script execution
5. Post-installation reload

Within these phases, other process and decision points occur. The following illustration shows a flow diagram of the POAP process.

Figure 9: POAP Process


Power up switch.
Startup configuration exists?
Yes
No
Bootup normally with startup configuration.
Abort POAP process?
Yes
No
Begin existing interactive setup over serial console.
Execute DHCP discovery and obtain IP address and TFTP server address of where to get the POAP script file.
Switch downloads the POAP script file and executes it.
Does the switch bootflash contain the image noted in the script file?
No
Switch downloads the image noted in the script.
Yes
Switch determines the name of the configuration file and downloads it.
Switch reboots.
Switch replays the configuration file to configure the switch.
Configuration is saved locally to NVRAM.
332315

Power-Up Phase

When you power up a switch for the first time, it loads the software image that is installed at manufacturing and tries to find a configuration file from which to boot. When a configuration file is not found, POAP mode starts.

During startup, a prompt appears asking if you want to abort POAP and continue with a normal setup. You can choose to exit or continue with POAP.

Note No user intervention is required for POAP to continue. The prompt that asks if you want to abort POAP remains available until the POAP process is complete.

If you exit POAP mode, you enter the normal interactive setup script. If you continue in POAP mode, all the front-panel interfaces are set up in the default configuration.

DHCP Discovery Phase

The switch sends out DHCP discover messages on the MGMT interface that solicits DHCP offers from the DHCP server or servers. (See the following figure.) The DHCP client on the Cisco Nexus switch uses the switch serial number in the client-identifier option to identify itself to the DHCP server. The DHCP server can use this identifier to send information, such as the IP address and script filename, back to the DHCP client.

POAP requires a minimum DHCP lease period of 3600 seconds (1 hour). POAP checks the DHCP lease period. If the DHCP lease period is set to less than 3600 seconds (1 hour), POAP does not complete the DHCP negotiation.

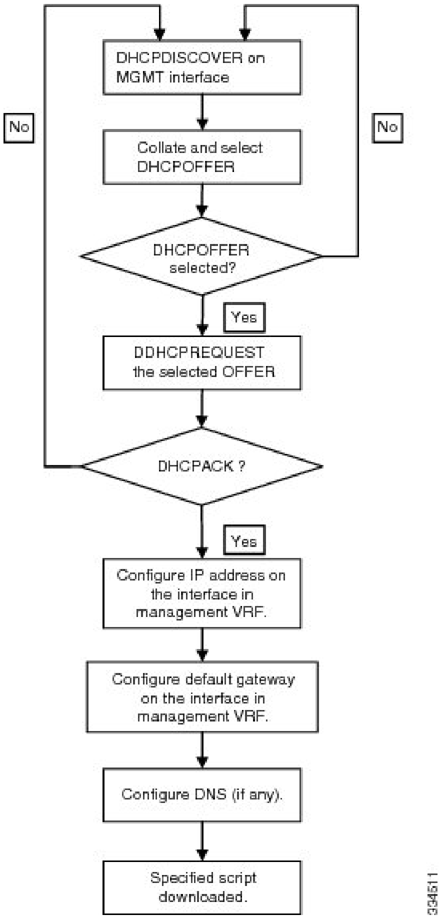
The DHCP discover message also solicits the following options from the DHCP server.

- TFTP server name or TFTP server address—The DHCP server relays the TFTP server name or TFTP server address to the DHCP client. The DHCP client uses this information to contact the TFTP server to obtain the script file.
- Bootfile name—The DHCP server relays the bootfile name to the DHCP client. The bootfile name includes the complete path to the bootfile on the TFTP server. The DHCP client uses this information to download the script file.

When multiple DHCP offers that meet the requirement are received, an offer is randomly chosen. The device completes the DHCP negotiation (request and acknowledgment) with the selected DHCP server, and the DHCP server assigns an IP address to the switch. If a failure occurs in any of the subsequent steps in the POAP process, the IP address is released back to the DHCP server.

If no DHCP offers meet the requirements, the switch does not complete the DHCP negotiation (request and acknowledgment) and an IP address is not assigned.

Figure 10: DHCP Discovery Process


DHCPDISCOVER on MGMT interface
No
No
Collate and select DHCPOFFER
DHCPOFFER selected?
Yes
DDHCPREQUEST the selected OFFER
DHCPACK ?
Yes
Configure IP address on the interface in management VRF.
Configure default gateway on the interface in management VRF.
Configure DNS (if any).
Specified script downloaded.
334511

Script Execution Phase

After the device bootstraps itself using the information in the DHCP acknowledgement, the script file is downloaded from the TFTP server.

The switch runs the configuration script, which downloads and installs the software image and downloads a switch-specific configuration file.

However, the configuration file is not applied to the switch at this point, because the software image that currently runs on the switch might not support all of the commands in the configuration file. After the switch reboots, it begins running the new software image, if an image was installed. At that point, the configuration is applied to the switch.

Note If the switch loses connectivity, the script stops, and the switch reloads its original software images and bootup variables.

Post-Installation Reload Phase

The switch restarts and applies (replays) the configuration on the upgraded software image. Afterward, the switch copies the running configuration to the startup configuration.

Setting Up the Network Environment to Use POAP

Procedure

Step 1 Modify the basic configuration script provided by Cisco or create your own script.

Step 2 (Optional) Put the POAP configuration script and any other desired software image and switch configuration files on a USB device that is accessible to the switch.

Step 3 Deploy a DHCP server and configure it with the interface, gateway, and TFTP server IP addresses and a bootfile with the path and name of the configuration script file. (This information is provided to the switch when it first boots.)

You do not need to deploy a DHCP server if all software image and switch configuration files are on the USB device.

Step 4 Deploy a TFTP server to host the configuration script.

Step 5 Deploy one or more servers to host the software images and configuration files.

Configuring a Switch Using POAP

Before you begin

Make sure that the network environment is set up to use POAP. For more information, refer to the "Setting up the Network Enviraonment to use POAP" section immediately preceding this section.

Procedure

Step 1 Install the switch in the network.

Step 2 Power on the switch.

If no configuration file is found, the switch boots in POAP mode and displays a prompt that asks if you want to abort POAP and continue with a normal setup.

No entry is required to continue to boot in POAP mode.

Step 3 (Optional) If you want to exit POAP mode and enter the normal interactive setup script, enter **y** (yes).

The switch boots, and the POAP process begins. For more information, see the "POAP Process" section.

What to do next

Verify the configuration.

Verifying the Device Configuration

To verify the configuration after bootstrapping the device using POAP, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.

For detailed information about the fields in the output from these commands, see the Cisco Nexus command reference for your device.

CHAPTER 8

Understanding the Command-Line Interface

This chapter helps you understand the command-line interface.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Information About the CLI Prompt

Once you have successfully accessed the device, the CLI prompt displays in the terminal window of your console port or remote workstation as shown in this example:

```
User Access Verification
login: admin
Password:<password>
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2009, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch#
```

You can change the default device hostname.

From the CLI prompt, you can do the following:

- Use CLI commands for configuring features
- Access the command history
- Use command parsing functions

Note In normal operation, usernames are case sensitive. However, when you are connected to the device through its console port, you can enter a login username in all uppercase letters regardless of how the username was defined. As long as you provide the correct password, the device logs you in.

Command Modes

This section describes command modes in the Cisco NX-OS CLI.

EXEC Command Mode

When you first log in, the Cisco NX-OS software places you in EXEC mode. The commands available in EXEC mode include the **show** commands that display the device status and configuration information, the **clear** commands, and other commands that perform actions that you do not save in the device configuration.

Global Configuration Command Mode

Global configuration mode provides access to the broadest range of commands. The term indicates characteristics or features that affect the device as a whole. You can enter commands in global configuration mode to configure your device globally or to enter more specific configuration modes to configure specific elements such as interfaces or protocols.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.
		Note The CLI prompt changes to indicate that you are in global configuration mode.

Interface Configuration Command Mode

One example of a specific configuration mode that you enter from global configuration mode is interface configuration mode. To configure interfaces on your device, you must specify the interface and enter interface configuration mode.

You must enable many features on a per-interface basis. Interface configuration commands modify the operation of the interfaces on the device, such as Ethernet interfaces or management interfaces (mgmt 0).

For more information about configuring interfaces, see the Cisco Nexus interfaces guide for your device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type number* **Example:** switch(config)# interface ethernet 2/2 switch(config-if)#	Specifies the interface that you want to configure. The CLI places you into interface configuration mode for the specified interface. **Note** The CLI prompt changes to indicate that you are in interface configuration mode.

Subinterface Configuration Command Mode

From global configuration mode, you can access a configuration submode for configuring VLAN interfaces called subinterfaces. In subinterface configuration mode, you can configure multiple virtual interfaces on a single physical interface. Subinterfaces appear to a protocol as distinct physical interfaces.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, you can configure IEEE 802.1Q encapsulation to associate a subinterface with a VLAN.

For more information about configuring subinterfaces, see the Cisco Nexus interfaces guide for your device. For details about the subinterface commands, see the command reference guide for your device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type number***.***subint*
	Example:
	`switch(config)# interface ethernet 2/2.1`
	`switch(config-subif)#`
		Specifies the VLAN interface to be configured.
		The CLI places you into a subinterface configuration mode for the specified VLAN interface.
		Note The CLI prompt changes to indicate that you are in global configuration mode.

Saving and Restoring a Command Mode

The Cisco NX-OS software allows you to save the current command mode, configure a feature, and then restore the previous command mode. The **push** command saves the command mode and the **pop** command restores the command mode.

This example shows how to save and restore a command mode:

```
switch# configure terminal
switch(config)# event manager applet test
switch(config-applet)# push
switch(config-applet)# configure terminal
switch(config)# username testuser password newtest
switch(config)# pop
switch(config-applet)#
```

Exiting a Configuration Command Mode

To exit from any configuration command mode, perform one of the following tasks:

Procedure

	Command or Action	Purpose

Step 1	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits from the current configuration command mode and returns to the previous configuration command mode.

Step 2	**end**
	Example:
	`switch(config-if)# end`
	`switch#`
		Exits from the current configuration command mode and returns to EXEC mode.

Step 3	(Optional) **Ctrl-Z**
	Example:
	`switch(config-if)# ^Z`
	`switch#`
		Exits the current configuration command mode and returns to EXEC mode.
		Caution If you press **Ctrl-Z** at the end of a command line in which a valid command has been typed, the CLI adds the command to the running configuration file. In most cases, you should exit a configuration mode using the **exit** or **end** command.

Command Mode Summary

This table summarizes information about the main command modes.

Table 7: Command Mode Summary

Mode	Access Method	Prompt	Exit Method
EXEC	From the login prompt, enter your username and password.	`switch#`	To exit to the login prompt, use the **exit** command.
Global configuration	From EXEC mode, use the **configure terminal** command.	`switch(config)#`	To exit to EXEC mode, use the **end** or **exit** command or press **Ctrl-Z**.
Interface configuration	From global configuration mode, use an interface command and specify an interface with an **interface** command.	`switch(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit to EXEC mode, use the **exit** command or press **Ctrl-Z**.

Mode	Access Method	Prompt	Exit Method
Subinterface configuration	From global configuration mode, specify a subinterface with an **interface** command.	`switch(config-subif)#`	To exit to global configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.

Special Characters

This table lists the characters that have special meaning in Cisco NX-OS text strings and should be used only in regular expressions or other special contexts.

Table 8: Special Characters

Character	Description
%	Percent
#	Pound, hash, or number
...	Ellipsis
|	Vertical bar
< >	Less than or greater than
[ ]	Brackets
{ }	Braces

Keystroke Shortcuts

This table lists command key combinations that can be used in both EXEC and configuration modes.

Table 9: Keystroke Shortcuts

Keystokes	Description
Ctrl-A	Moves the cursor to the beginning of the line.
Ctrl-B	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Ctrl-C	Cancels the command and returns to the command prompt.
Ctrl-D	Deletes the character at the cursor.

Keystokes	Description
Ctrl-E	Moves the cursor to the end of the line.
Ctrl-F	Moves the cursor one character to the right.
Ctrl-G	Exits to the previous command mode without removing the command string.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L	Redisplays the current command line.
Ctrl-N	Displays the next command in the command history.
Ctrl-O	Clears the terminal screen.
Ctrl-P	Displays the previous command in the command history.
Ctrl-R	Redisplays the current command line.
Ctrl-T	Transposes the character under the cursor with the character located to the right of the cursor. The cursor is then moved one character to the right.
Ctrl-U	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-V	Removes any special meaning for the following keystroke. For example, press Ctrl-V before entering a question mark (?) in a regular expression.
Ctrl-W	Deletes the word to the left of the cursor.
Ctrl-X, H	Lists the history of commands you have entered. When using this key combination, press and release the Ctrl and X keys together before pressing H.
Ctrl-Y	Recalls the most recent entry in the buffer (press keys simultaneously).
Ctrl-Z	Ends a configuration session, and returns you to EXEC mode. When used at the end of a command line in which a valid command has been typed, the resulting configuration is first added to the running configuration file.
Up arrow key	Displays the previous command in the command history.
Down arrow key	Displays the next command in the command history.
Right arrow key Left arrow key	Moves your cursor through the command string, either forward or backward, allowing you to edit the current command.
?	Displays a list of available commands.

<table>
<tr><th>Keystokes</th><th>Description</th></tr>
<tr><td rowspan="4">Tab</td><td>Completes the word for you after you enter the first characters of the word and then press the Tab key. All options that match are presented.
Use tabs to complete the following items:
• Command names
• Scheme names in the file system
• Server names in the file system
• Filenames in the file system</td></tr>
<tr><td><b>Example:</b><pre>
switch(config)# <b>c<Tab></b>
callhome class-map clock cts
cdp   cli   control-plane
switch(config)# <b>cl<Tab></b>
class-map   cli   clock
switch(config)# <b>cla<Tab></b>
switch(config)# <b>class-map</b>
</pre></td></tr>
<tr><td><b>Example:</b><pre>
switch# <b>cd bootflash:<Tab></b>
bootflash: bootflash://sup-1/
bootflash:///   bootflash://sup-2/
bootflash://module-5/ bootflash://sup-active/
bootflash://module-6/ bootflash://sup-local/
</pre></td></tr>
<tr><td><b>Example:</b><pre>
switch# <b>cd bootflash://mo<Tab></b>
bootflash://module-5/ bootflash://module-6/cv
switch# <b>cd bootflash://module-</b>
</pre></td></tr>
</table>

Abbreviating Commands

You can abbreviate commands and keywords by entering the first few characters of a command. The abbreviation must include sufficient characters to make it unique from other commands or keywords. If you are having trouble entering a command, check the system prompt and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

This table lists examples of command abbreviations.

Table 10: Examples of Command Abbreviations

Command	Abbreviation
configure terminal	**conf t**

Command	Abbreviation
copy running-config startup-config	**copy run start**
interface ethernet 1/2	**int e 1/2**
show running-config	**sh run**

Completing a Partial Command Name

If you cannot remember a complete command name, or if you want to reduce the amount of typing you have to perform, enter the first few letters of the command, and then press the **Tab** key. The command line parser will complete the command if the string entered is unique to the command mode. If your keyboard does not have a **Tab** key, press **Ctrl-I** instead.

The CLI recognizes a command once you have entered enough characters to make the command unique. For example, if you enter **conf** in EXEC mode, the CLI will be able to associate your entry with the **configure** command, because only the **configure** command begins with **conf**.

In this example, the CLI recognizes the unique string for **conf** in EXEC mode when you press the **Tab** key:

```
switch# conf<Tab>
switch# configure
```

When you use the command completion feature the CLI displays the full command name. The CLI does not execute the command until you press the **Return** or **Enter** key. This feature allows you to modify the command if the full command was not what you intended by the abbreviation. If you enter a set of characters that could indicate more than one command, a list of matching commands displays.

For example, entering **co<Tab>** lists all commands available in EXEC mode beginning with **co**:

```
switch# co<Tab>
configure   copy
switch# co
```

Note that the characters you entered appear at the prompt again to allow you to complete the command entry.

Identifying Your Location in the Command Hierarchy

Some features have a configuration submode hierarchy nested more than one level. In these cases, you can display information about your present working context (PWC).

Procedure

	Command or Action	Purpose
Step 1	**where detail** **Example:** `switch# configure terminal`	Displays the PWC.

	Command or Action	Purpose

	`switch(config)# interface mgmt0`
	`switch(config-if)# where detail`
	`mode:                conf`
	`                        interface mgmt0`
	`  username:            admin`	

Using the no Form of a Command

Almost every configuration command has a **no** form that can be used to disable a feature, revert to a default value, or remove a configuration. The Cisco NX-OS command reference publications describe the function of the **no** form of the command whenever a **no** form is available.

This example shows how to disable a feature:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# no feature tacacs+
```

This example shows how to revert to the default value for a feature:

```
switch# configure terminal
switch(config)# banner motd #Welcome to the switch#
switch(config)# show banner motd
Welcome to the switch

switch(config)# no banner motd
switch(config)# show banner motd
User Access Verification
```

This example shows how to remove the configuration for a feature:

```
switch# configure terminal
switch(config)# radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1

following RADIUS servers are configured:
        10.10.1.1:
                available for authentication on port:1812
                available for accounting on port:1813
        10.10.2.2:
                available for authentication on port:1812
                available for accounting on port:1813

switch(config)# no radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1
```

```
following RADIUS servers are configured:
        10.10.1.1:
                available for authentication on port:1812
                available for accounting on port:1813
```

This example shows how to use the **no** form of a command in EXEC mode:

```
switch# cli var name testinterface ethernet1/2
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2009-05-12-13.43.13"
testinterface="ethernet1/2"

switch# cli no var name testinterface
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2009-05-12-13.43.13"
```

Configuring CLI Variables

This section describes CLI variables in the Cisco NX-OS CLI.

About CLI Variables

The Cisco NX-OS software supports the definition and use of variables in CLI commands.

You can refer to CLI variables in the following ways:

- Entered directly on the command line.
- Passed to a script initiated using the **run-script** command. The variables defined in the parent shell are available for use in the child **run-script** command process.

CLI variables have the following characteristics:

- Cannot have nested references through another variable
- Can persist across switch reloads or exist only for the current session

Cisco NX-OS supports one predefined variable: TIMESTAMP. This variable refers to the current time when the command executes in the format YYYY-MM-DD-HH.MM.SS.

Note The TIMESTAMP variable name is case sensitive. All letters must be uppercase.

Configuring CLI Session-Only Variables

You can define CLI session variables to persist only for the duration of your CLI session. These variables are useful for scripts that you execute periodically. You can reference the variable by enclosing the name in parentheses and preceding it with a dollar sign ($), for example $(*variable-name*).

Procedure

	Command or Action	Purpose
Step 1	**cli var name** *variable-name variable-text* **Example:** `switch# cli var name testinterface ethernet 2/1`	Configures the CLI session variable. The *variable-name* argument is alphanumeric, case sensitive, and has a maximum length of 31 characters. The *variable-text* argument is alphanumeric, case sensitive, can contain spaces, and has a maximum length of 200 characters.
Step 2	(Optional) **show cli variables** **Example:** `switch# show cli variables`	Displays the CLI variable configuration.

Configuring Persistent CLI Variables

You can configure CLI variables that persist across CLI sessions and device reloads.

Procedure

	Command or Action	Purpose

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

Step 2	**cli var name** *variable-name variable-text* **Example:** `switch(config)# cli var name testinterface ethernet 2/1`	Configures the CLI persistent variable. The variable name is a case-sensitive, alphanumeric string and must begin with an alphabetic character. The maximum length is 31 characters.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show cli variables** **Example:** `switch# show cli variables`	Displays the CLI variable configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Command Aliases

This section provides information about command aliases.

About Command Aliases

You can define command aliases to replace frequently used commands. The command aliases can represent all or part of the command syntax.

Command alias support has the following characteristics:

- Command aliases are global for all user sessions.
- Command aliases persist across reboots if you save them to the startup configuration.
- Command alias translation always takes precedence over any keyword in any configuration mode or submode.
- Command alias configuration takes effect for other user sessions immediately.
- The Cisco NX-OS software provides one default alias, **alias**, which is the equivalent to the **show cli alias** command that displays all user-defined aliases.
- You cannot delete or change the default command alias **alias**.
- You can nest aliases to a maximum depth of 1. One command alias can refer to another command alias that must refer to a valid command, not to another command alias.
- A command alias always replaces the first command keyword on the command line.
- You can define command aliases for commands in any command mode.
- If you reference a CLI variable in a command alias, the current value of the variable appears in the alias, not the variable reference.
- You can use command aliases for **show** command searching and filtering.

Defining Command Aliases

You can define command aliases for commonly used commands.

Procedure

	Command or Action	Purpose

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

Step 2	**cli alias name** *alias-name alias-text* **Example:**	Configures the command alias. The alias name is an alphanumeric string that is not case

	Command or Action	Purpose
	`switch(config)# cli alias name ethint interface ethernet`	sensitive and must begin with an alphabetic character. The maximum length is 30 characters.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **alias** **Example:** `switch# alias`	Displays the command alias configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Command Aliases for a User Session

You can create a command alias for the current user session that is not available to any other user on the Cisco NX-OS device. You can also save the command alias for future use by the current user account.

Procedure

	Command or Action	Purpose
Step 1	**terminal alias** [**persist**] *alias-name command -string* **Example:** `switch# terminal alias shintbr show interface brief`	Configures a command alias for the current user session. Use the **persist** keyword to save the alias for future use by the user account. **Note** Do not abbreviate the **persist** keyword.

Command Scripts

This section describes how you can create scripts of commands to perform multiple tasks.

Running a Command Script

You can create a list of commands in a file and execute them from the CLI. You can use CLI variables in the command script.

Note You cannot create the script files at the CLI prompt. You can create the script file on a remote device and copy it to the bootflash:, slot0:, or volatile: directory on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**run-script** [**bootflash:** | **slot0:** | **volatile:**]*filename* **Example:** `switch# run-script testfile`	Executes the commands in the file on the default directory.

Echoing Information to the Terminal

You can echo information to the terminal, which is particularly useful from a command script. You can reference CLI variables and use formatting options in the echoed text.

This table lists the formatting options that you can insert in the text.

Table 11: Formatting Options for the echo Command

Formatting Option	Description
\b	Inserts back spaces.
\c	Removes the new line character at the end of the text string.
\f	Inserts a form feed character.
\n	Inserts a new line character.
\r	Returns to the beginning of the text line.
\t	Inserts a horizontal tab character.
\v	Inserts a vertical tab character.
\\\\	Displays a backslash character.
\nnn	Displays the corresponding ASCII octal character.

Procedure

	Command or Action	Purpose

Step 1	**echo** [**backslash-interpret**] [*text*]
	Example:
	`switch# echo This is a test.`
	`This is a test.`
		The **backslash-interpret** keyword indicates that the text string contains formatting options. The *text* argument is alphanumeric, case sensitive, and can contain blanks. The maximum length is 200 characters. The default is a blank line.

Delaying Command Action

You can delay a command action for a period of time, which is particularly useful within a command script.

Procedure

	Command or Action	Purpose
Step 1	**sleep** *seconds* **Example:** switch# sleep 30	Causes a delay for a number of seconds. The range is from 0 to 2147483647.

Context-Sensitive Help

The Cisco NX-OS software provides context-sensitive help in the CLI. You can use a question mark (?) at any point in a command to list the valid input options.

CLI uses the caret (^) symbol to isolate input errors. The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument.

This table shows example outputs of context sensitive help.

Table 12: Context-Sensitive Help Example

Example Outputs	Description
switch# **clock ?** set HH:MM:SS Current Time switch# clock	Displays the command syntax for the **clock** command in EXEC mode. The switch output shows that the **set** keyword is required for using the **clock** command.
switch# clock **set ?** WORD HH:MM:SS Current Time switch# clock set	Displays the command syntax for setting the time. The help output shows that the current time is required for setting the clock and how to format the time.
switch# clock set **13:32:00<CR>** % Incomplete command switch#	Adds the current time. The CLI indicates the command is incomplete.
switch# **<Ctrl-P>** switch# clock set 13:32:00	Displays the previous command that you entered.
switch# clock set 13:32:00 **?** <1-31> Day of the month switch# clock set 13:32:00	Displays the additional arguments for the **clock set** command.

Example Outputs	Description

`switch# clock set 13:32:00 18 ?`
`  April      Month of the year`
`  August     Month of the year`
`  December   Month of the year`
`  February   Month of the year`
`  January    Month of the year`
`  July       Month of the year`
`  June       Month of the year`
`  March      Month of the year`
`  May        Month of the year`
`  November   Month of the year`
`  October    Month of the year`
`  September  Month of the year`
`switch# clock set 13:32:00 18`
	Displays the additional arguments for the **clock set** command.

`switch# clock set 13:32:00 18 April 08<CR>`
`% Invalid input detected at '^' marker.`
	Adds the date to the clock setting.
	The CLI indicates an error with the caret symbol (^) at 08.

`switch# clock set 13:32:00 18 April ?`
`  <2000-2030>  Enter the year (no abbreviation)`
 	
`switch# clock set 13:32:00 18 April`
	Displays the correct arguments for the year.

`switch# clock set 13:32:00 18 April 2008<CR>`
`switch#`
	Enters the correct syntax for the **clock set** command.

Understanding Regular Expressions

The Cisco NX-OS software supports regular expressions for searching and filtering in CLI output, such as the **show** commands. Regular expressions are case sensitive and allow for complex matching requirements.

Special Characters

You can also use other keyboard characters (such as ! or ~) as single-character patterns, but certain keyboard characters have special meanings when used in regular expressions.

This table lists the keyboard characters that have special meanings.

Table 13: Special Characters with Special Meaning

Character	Special Meaning
.	Matches any single character, including white space.
*	Matches 0 or more sequences of the pattern.
+	Matches 1 or more sequences of the pattern.
?	Matches 0 or 1 occurrences of the pattern.

Character	Special Meaning
^	Matches the beginning of the string.
$	Matches the end of the string.
_ (underscore)	Matches a comma (,), left brace ({), right brace (}), left parenthesis (( ), right parenthesis ( )), the beginning of the string, the end of the string, or a space. **Note** The underscore is only treated as a regular expression for BGP related commands.

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\). This example contains single-character patterns that match a dollar sign ($), an underscore (_), and a plus sign (+), respectively:

```
\$ \_ \+
```

Multiple-Character Patterns

You can also specify a pattern that contains multiple characters by joining letters, digits, or keyboard characters that do not have special meanings. For example, a4% is a multiple-character regular expression.

With multiple-character patterns, the order is important. The regular expression **`a4%`** matches the character a followed by a 4 followed by a percent sign (%). If the string does not have a4%, in that order, pattern matching fails. The multiple-character regular expression **`a.`** (the character a followed by a period) uses the special meaning of the period character to match the letter a followed by any single character. With this example, the strings ab, a!, or a2 are all valid matches for the regular expression.

You can remove the special meaning of a special character by inserting a backslash before it. For example, when the expression **`a\.`** is used in the command syntax, only the string a. will be matched.

Anchoring

You can match a regular expression pattern against the beginning or the end of the string by anchoring these regular expressions to a portion of the string using the special characters.

This table lists the special characters that you can use for anchoring.

Table 14: Special Characters Used for Anchoring

Character	Description
^	Matches the beginning of the string.
$	Matches the end of the string.

For example, the regular expression **^con** matches any string that starts with **con**, and **sole$** matches any string that ends with **sole**.

Note The ^ symbol can also be used to indicate the logical function "not" when used in a bracketed range. For example, the expression **[^abcd]** indicates a range that matches any single letter, as long as it is not a, b, c, or d.

Searching and Filtering show Command Output

Often, the output from **show** commands can be lengthy and cumbersome. The Cisco NX-OS software provides the means to search and filter the output so that you can easily locate information. The searching and filtering options follow a pipe character (|) at the end of the **show** command. You can display the options using the CLI context-sensitive help facility:

```
switch# show running-config | ?
  cut      Print selected parts of lines.
  diff     Show difference between current and previous invocation (creates temp files:
           remove them with 'diff-clean' command and don't use it on commands with big
           outputs, like 'show tech'!)
  egrep    Egrep - print lines matching a pattern
  grep     Grep - print lines matching a pattern
  head     Display first lines
  human    Output in human format
  last     Display last lines
  less     Filter for paging
  no-more  Turn-off pagination for command output
  perl     Use perl script to filter output
  section  Show lines that include the pattern as well as the subsequent lines that are
           more indented than matching line
  sed      Stream Editor
  sort     Stream Sorter
  sscp     Stream SCP (secure copy)
  tr       Translate, squeeze, and/or delete characters
  uniq     Discard all but one of successive identical lines
  vsh      The shell that understands cli command
  wc       Count words, lines, characters
  begin    Begin with the line that matches
  count    Count number of lines
  end      End with the line that matches
  exclude  Exclude lines that match
  include  Include lines that match
```

Filtering and Searching Keywords

The Cisco NX-OS CLI provides a set of keywords that you can use with the **show** commands to search and filter the command output.

This table lists the keywords for filtering and searching the CLI output.

Table 15: Filtering and Searching Keywords

Keyword Syntax	Description

begin *string*
Example:
`show version | begin Hardware`
	Starts displaying at the line that contains the text that matches the search string. The search string is case sensitive.

count **Example:** `show running-config | count`	Displays the number of lines in the command output.

cut [**-d** *character*] {**-b** | **-c** | **-f** | **-s**}
Example:
`show file testoutput | cut -b 1-10`
	Displays only part of the output lines. You can display a number of bytes (**-b**), characters (**-vcut** [**-d** *character*] {**-b** | **-c** | **-f** | **-s**}), or fields (**-f**). You can also use the **-d** keyword to define a field delimiter other than the tag character default. The **-s** keyword suppresses the display of the lines that do not contain the delimiter.

end *string*
Example:
`show running-config | end interface`
	Displays all lines up to the last occurrence of the search string.

exclude *string* **Example:** `show interface brief | exclude down`	Displays all lines that do not include the search string. The search string is case sensitive.
head [**lines** *lines*] **Example:** `show logging logfile | head lines 50`	Displays the beginning of the output for the number of lines specified. The default number of lines is 10.
include *string* **Example:** `show interface brief | include up`	Displays all lines that include the search string. The search string is case sensitive.
last [*lines*] **Example:** `show logging logfile | last 50`	Displays the end of the output for the number of lines specified. The default number of lines is 10.
no-more **Example:** `show interface brief | no-more`	Displays all the output without stopping at the end of the screen with the `--More--` prompt.

Keyword Syntax	Description
sscp *SSH-connection-name filename* **Example:** `show version | sscp MyConnection show_version_output`	Redirects the output using streaming secure copy (sscp) to a named SSH connection. You can create the SSH named connection using the **ssh name** command.
wc [**bytes** | **lines** | **words**] **Example:** `show file testoutput | wc bytes`	Displays counts of characters, lines, or words. The default is to display the number of lines, words, and characters.

diff Utility

You can compare the output from a **show** command with the output from the previous invocation of that command.

diff-clean [**all-session**] [**all-users**]

This table describes the keywords for the diff utility.

Keyword	Description
all-sessions	Removes diff temporary files from all sessions (past and present sessions) of the current user.
all-users	Removes diff temporary files from all sessions (past and present sessions) of all users.

The Cisco NX-OS software creates temporary files for the most current output for a **show** command for all current and previous users sessions. You can remove these temporary files using the **diff-clean** command.

diff-clean [**all-sessions** | **all-users**]

By default, the **diff-clean** command removes the temporary files for the current user's active session. The **all-sessions** keyword removes temporary files for all past and present sessions for the current user. The **all-users** keyword removes temporary files for all past and present sessions for the all users.

grep and egrep Utilities

You can use the Global Regular Expression Print (grep) and Extended grep (egrep) command-line utilities to filter the **show** command output.

The grep and egrep syntax is as follows:

{**grep** | **egrep**} [**count**] [**ignore-case**] [**invert-match**] [**line-exp**] [**line-number**] [**next** *lines*] [**prev** *lines*] [**word-exp**] *expression*}]

This table lists the **grep** and **egrep** parameters.

Table 16: grep and egrep Parameters

Parameter	Description
count	Displays only the total count of matched lines.
ignore-case	Specifies to ignore the case difference in matched lines.
invert-match	Displays lines that do not match the expression.
line-exp	Displays only lines that match a complete line.
line-number	Specifies to display the line number before each matched line.
next *lines*	Specifies the number of lines to display after a matched line. The default is 0. The range is from 1 to 999.
prev *lines*	Specifies the number of lines to display before a matched line. The default is 0. The range is from 1 to 999.
word-exp	Displays only lines that match a complete word.
expression	Specifies a regular expression for searching the output.

less Utility

You can use the less utility to display the contents of the **show** command output one screen at a time. You can enter **less** commands at the : prompt. To display all **less** commands you can use, enter **h** at the : prompt.

sed Utility

You can use the Stream Editor (sed) utility to filter and manipulate the **show** command output as follows:

sed *command*

The *command* argument contains sed utility commands.

sort Utility

You can use the sort utility to filter **show** command output.

The sort utility syntax is as follows:

sort [**-M**] [**-b**] [**-d**] [**-f**] [**-g**] [**-i**] [**-k** *field-number*[**.***char-position*][*ordering*]] [**-n**] [**-r**] [**-t** *delimiter*] [**-u**]

This table describes the sort utility parameters.

Table 17: sort Utility Parameters

Parameter	Description
-M	Sorts by month.

Parameter	Description
-b	Ignores leading blanks (space characters). The default sort includes the leading blanks.
-d	Sorts by comparing only blanks and alphanumeric characters. The default sort includes all characters.
-f	Folds lowercase characters into uppercase characters.
-g	Sorts by comparing a general numeric value.
-i	Sorts only using printable characters. The default sort includes nonprintable characters.
-k *field-number*[**.***char-position*][*ordering*]	Sorts according to a key value. There is no default key value.
-n	Sorts according to a numeric string value.
-r	Reverses order of the sort results. The default sort output is in ascending order.
-t *delimiter*	Sorts using a specified delimiter. The default delimiter is the space character.
-u	Removes duplicate lines from the sort results. The sort output displays the duplicate lines.

Searching and Filtering from the --More-- Prompt

You can search and filter output from `--More--` prompts in the **show** command output.

This table describes the `--More--` prompt commands.

Table 18: --More-- Prompt Commands

Commands	Description
[*lines*]<space>	Displays output lines for either the specified number of lines or the current screen size.
[*lines*]**z**	Displays output lines for either the specified number of lines or the current screen size. If you use the *lines* argument, that value becomes the new default screen size.
[*lines*]<return>	Displays output lines for either the specified number of lines or the current default number of lines. The initial default is 1 line. If you use the optional *lines* argument, that value becomes the new default number of lines to display for this command.

Commands	Description
[*lines*]**d** or [*lines*]Ctrl+shift+D	Scrolls through output lines for either the specified number of lines or the current default number of lines. The initial default is 11 lines. If you use the optional *lines* argument, that value becomes the new default number of lines to display for this command.
q or **Q** or Ctrl-C	Exits the `--More--` prompt.
[*lines*]**s**	Skips forward in the output for either the specified number of lines or the current default number of lines and displays a screen of lines. The default is 1 line.
[*lines*]**f**	Skips forward in the output for either the specified number of screens or the current default number of screens and displays a screen of lines. The default is 1 screen.
=	Displays the current line number.
[*count*]/*expression*	Skips to the line that matches the regular expression and displays a screen of output lines. Use the optional *count* argument to search for lines with multiple occurrences of the expression. This command sets the current regular expression that you can use in other commands.
[*count*]**n**	Skips to the next line that matches the current regular expression and displays a screen of output lines. Use the optional *count* argument to skip past matches.
{**!** | **:!**[*shell-cmd*]}	Executes the command specified in the *shell-cmd* argument in a subshell.
.	Repeats the previous command.

Using the Command History

The Cisco NX-OS software CLI allows you to access the command history for the current user session. You can recall and reissue commands, with or without modification. You can also clear the command history.

Recalling a Command

You can recall a command in the command history to optionally modify and enter again.

This example shows how to recall a command and reenter it:

```
switch(config)# show cli history
0  11:04:07   configure terminal
1  11:04:28   show interface ethernet 2/24
2  11:04:39     interface ethernet 2/24
3  11:05:13       no shutdown
4  11:05:19     exit
5  11:05:25   show cli history
switch(config)# !1
switch(config)# show interface ethernet 2/24
```

You can also use the **Ctrl-P** and **Ctrl-N** keystroke shortcuts to recall commands.

Controlling CLI History Recall

You can control the commands that you recall from the CLI history using the **Ctrl-P** and **Ctrl-N** keystroke shortcuts. Cisco NX-OS software recalls all commands from the current command mode and higher command modes. For example, if you are working in global configuration mode, the command recall keystroke shortcuts recall both EXEC mode and global configuration mode commands.

Configuring the CLI Edit Mode

You can recall commands from the CLI history using the **Ctrl-P** and **Ctrl-N** keystroke shortcuts and edit them before reissuing them. The default edit mode is emacs. You can change the edit mode to vi.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **terminal edit-mode vi** [**persist**] **Example:** `switch# terminal edit-mode vi`	Changes the CLI edit mode to vi for the user session. The **persist** keyword makes the setting persistent across sessions for the current username. Use the **no** to revert to using emacs.

Displaying the Command History

You can display the command history using the **show cli history** command.

The **show cli history** command has the following syntax:

By default, the number of lines displayed is 12 and the output includes the command number and timestamp.

The example shows how to display default number of lines of the command history:

```
switch# show cli history
```

The example shows how to display 20 lines of the command history:

```
switch# show cli history 20
```

The example shows how to display only the commands in the command history without the command number and timestamp:

```
switch(config)# show cli history unformatted
```

Enabling or Disabling the CLI Confirmation Prompts

For many features, the Cisco NX-OS software displays prompts on the CLI that ask for confirmation before continuing. You can enable or disable these prompts. The default is enabled.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **terminal dont-ask** [**persist**] **Example:** `switch# terminal dont-ask`	Disables the CLI confirmation prompt. The **persist** keyword makes the setting persistent across sessions for the current username. The default is enabled. Use the **no** form of the command to enable the CLI confirmation prompts.

Setting CLI Display Colors

You can change the CLI colors to display as follows:

- The prompt displays in green if the previous command succeeded.
- The prompt displays in red of the previous command failed.
- The user input displays in blue.
- The command output displays in the default color.

The default colors are those set by the terminal emulator software.

Procedure

	Command or Action	Purpose
Step 1	**terminal color** [**evening**] [**persist**] **Example:** `switch# terminal color`	Sets the CLI display colors for the terminal session. The **evening** keyword is not supported. The **persist** keyword makes the setting persistent across sessions for the current username. The default setting is not persistent.

Sending Commands to Modules

You can send commands directly to modules from the supervisor module session using the **slot** command.

The **slot** has the following syntax:

slot *slot-number* [**quoted**] *command-string*

By default, the keyword and arguments in the *command-string* argument are separated by a space. To send more than one command to a module, separate the commands with a space character, a semicolon character (;), and a space character.

The**quoted** keyword indicates that the command string begins and ends with double quotation marks ("). Use this keyword when you want to redirect the module command output to a filtering utility, such as diff, that is supported only on the supervisor module session.

This example shows how to display and filter module information:

```
switch# slot 2 show version | grep lc
```

This example shows how to filter module information on the supervisor module session:

```
switch# slot 2 quoted "show version" | diff
switch# slot 4 quoted "show version" | diff -c
*** /volatile/vsh_diff_1_root_8430_slot__quoted_show_version.old       Wed Apr 29 20:10:41
 2009
--- -   Wed Apr 29 20:10:41 2009
***************
*** 1,5 ****
! RAM 1036860 kB
! lc2
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
--- 1,5 ----
! RAM 516692 kB
! lc4
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
***************
*** 12,16 ****
  Hardware
      bootflash: 0 blocks (block size 512b)

!    uptime is 0 days 1 hours 45 minute(s) 34 second(s)

--- 12,16 ----
  Hardware
      bootflash: 0 blocks (block size 512b)

!    uptime is 0 days 1 hours 45 minute(s) 42 second(s)
```

BIOS Loader Prompt

When the supervisor modules power up, a specialized BIOS image automatically loads and tries to locate a valid kickstart image for booting the system. If a valid kickstart image is not found, the following BIOS loader prompt displays:

```
loader>
```

For information on how to load the Cisco NX-OS software from the `<loader>` prompt, see the Cisco Nexus troubleshooting guide for your device.

Examples Using the CLI

This section includes examples of using the CLI.

Defining Command Aliases

This example shows how to define command aliases:

```
cli alias name ethint interface ethernet
cli alias name shintbr show interface brief
cli alias name shintupbr shintbr | include up | include ethernet
```

This example shows how to use a command alias:

```
switch# configure terminal
switch(config)# ethint 2/3
switch(config-if)#
```

Using CLI Session Variables

You can reference a variable using the syntax **$(**_variable-name_**)**.

This example shows how to reference a user-defined CLI session variable:

```
switch# show interface $(testinterface)
Ethernet2/1 is down (Administratively down)
  Hardware is 10/100/1000 Ethernet, address is 0000.0000.0000 (bia 0019.076c.4dac)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Switchport monitor is off
  Last clearing of "show interface" counters never
  5 minute input rate 0 bytes/sec, 0 packets/sec
  5 minute output rate 0 bytes/sec, 0 packets/sec
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause 0 reset
```

Using the System-Defined Timestamp Variable

This example uses $(TIMESTAMP) when redirecting **show** command output to a file:

```
switch# show running-config > rcfg.$(TIMESTAMP)
Preparing to copy....done
switch# dir
       12667     May 01 12:27:59 2008  rcfg.2008-05-01-12.27.59

Usage for bootflash://sup-local
8192 bytes used
20963328 bytes free
20971520 bytes total
```

Running a Command Script

This example displays the CLI commands specified in the script file:

```
switch# show file testfile
configure terminal
interface ethernet 2/1
no shutdown
end
show interface ethernet 2/1
```

This example displays the **run-script** command execution output:

```
switch# run-script testfile
`configure terminal`
`interface ethernet 2/1`
`no shutdown`
`end`
`show interface ethernet 2/1 `
Ethernet2/1 is down (Link not connected)
  Hardware is 10/100/1000 Ethernet, address is 0019.076c.4dac (bia 0019.076c.4dac)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is trunk
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Switchport monitor is off
  Last clearing of "show interface" counters 1d26.2uh
  5 minute input rate 0 bytes/sec, 0 packets/sec
  5 minute output rate 0 bytes/sec, 0 packets/sec
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
```

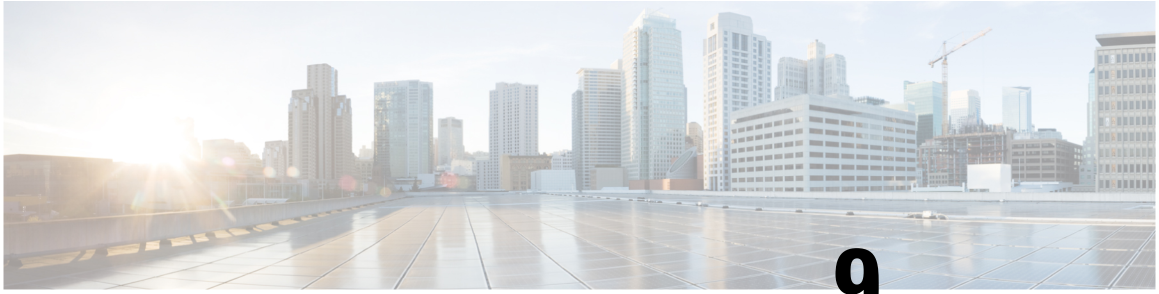
```
0 bad proto drop 0 if down drop 0 input with dribble
0 input discard
0 output error 0 collision 0 deferred
0 late collision 0 lost carrier 0 no carrier
0 babble
0 Rx pause 0 Tx pause 0 reset
```

Additional References for the CLI

This section includes additional information related to the CLI.

Related Documents for the CLI

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*



CHAPTER 9

Configuring Terminal Settings and Sessions

This chapter describes how to configure terminal settings and sessions.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Information About Terminal Settings and Sessions

This section includes information about terminal settings and sessions.

Terminal Session Settings

The Cisco NX-OS software features allow you to manage the following characteristics of terminals:

Terminal type
Name used by Telnet when communicating with remote hosts

Length
Number of lines of command output displayed before pausing

Width
Number of characters displayed before wrapping the line

Inactive session timeout
Number of minutes that a session remains inactive before the device terminates it

Console Port

The console port is an asynchronous serial port that allows you to connect to the device for initial configuration through a standard RS-232 port with an RJ-45 connector. Any device connected to this port must be capable of asynchronous transmission. You can configure the following parameters for the console port:

Data bits
Specifies the number of bits in an 8-bit byte that is used for data.

Inactive session timeout
Specifies the number of minutes a session can be inactive before it is terminated.

Parity
Specifies the odd or even parity for error detection.

Speed
Specifies the transmission speed for the connection.

Stop bits
Specifies the stop bits for an asynchronous line.

Configure your terminal emulator with 9600 baud, 8 data bits, 1 stop bit, and no parity.

COM1 Port

A COM1 port is an RS-232 port with a DB-9 interface that enables you to connect to an external serial communication device such as a modem. You can configure the following parameters for the COM1 port:

Data bits
Specifies the number of bits in an 8-bit byte that is used for data.

Hardware flowcontrol
Enables the flow-control hardware.

Parity
Specifies the odd or even parity for error detection.

Speed
Specifies the transmission speed for the connection.

Stop bits
Specifies the stop bits for an asynchronous line.

Configure your terminal emulator with 9600 baud, 8 data bits, 1 stop bit, and no parity.

Virtual Terminals

You can use virtual terminal lines to connect to your Cisco NX-OS device. Secure Shell (SSH) and Telnet create virtual terminal sessions. You can configure an inactive session timeout and a maximum sessions limit for virtual terminals.

Modem Support

You can connect a modem to the COM1 or console ports only on the supervisor 1 module. The following modems were tested on devices running the Cisco NX-OS software:

- MultiTech MT2834BA
- Hayes Accura V.92

Note Do not connect a modem when the device is booting. Only connect the modem when the device is powered up.

The Cisco NX-OS software has the default initialization string (ATE0Q1&D2&C1S0=1\015) to detect connected modems. The default string is defined as follows:

AT
Attention

E0 (required)
No echo

Q1
Result code on

&D2
Normal data terminal ready (DTR) option

&C1
Enable tracking the state of the data carrier

S0=1
Pick up after one ring

\015 (required)
Carriage return in octal

Configuring the Console Port

You can set the following characteristics for the console port:

- Data bits
- Inactive session timeout
- Parity
- Speed
- Stop bits

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

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

Step 3	**databits** *bits* **Example:** `switch(config-console)# databits 7`	Configures the number of data bits per byte. The range is from 5 to 8. The default is 8.
Step 4	**exec-timeout** *minutes* **Example:** `switch(config-console)# exec-timeout 30`	Configures the timeout for an inactive session. The range is from 0 to 525600 minutes (8760 hours). A value of 0 minutes disables the session timeout. The default is 30 minutes.

Step 5	**parity** {**even** | **none** | **odd**}
	Example:
	`switch(config-console)# parity even`
		Configures the parity. The default is **none**.

Step 6	**speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**}
	Example:
	`switch(config-console)# speed 115200`
		Configures the transmit and receive speed. The default is **9600**.

Step 7	**stopbits** {**1** | **2**}
	Example:
	`switch(config-console)# stopbits 2`
		Configures the stop bits. The default is **1**.

Step 8	**exit**
	Example:
	`switch(config-console)# exit`
	`switch(config)#`
		Exits console configuration mode.

Step 9	(Optional) **show line console** **Example:** `switch(config)# show line console`	Displays the console settings.
Step 10	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

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

Configuring the COM1 Port

You can set the following characteristics for the COM1 port:

- Data bits
- Flow control on the hardware
- Parity
- Speed
- Stop bits

Before you begin

Log in to the console port or COM1 port.

Procedure

	Command or Action	Purpose

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

Step 2	**line com1**
	Example:
	`switch# line com1`
	`switch(config-com1)#`
		Enters COM1 configuration mode.

Step 3	**databits** *bits* **Example:** `switch(config-com1)# databits 7`	Configures the number of data bits per byte. The range is from 5 to 8. The default is 8.
Step 4	**flowcontrol hardware** **Example:** `switch(config-com1)# flowcontrol hardware`	Enables flow control on the hardware. The default is enabled. Use the **no flowcontrol hardware** command to disable flow control on the hardware.

Step 5	**parity** {**even** | **none** | **odd**}
	Example:
	`switch(config-com1)# parity even`
		Configures the parity. The default is **none**.

	Command or Action	Purpose

Step 6	**speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**}
	Example:
	`switch(config-com1)# speed 115200`
		Configures the transmit and receive speed. The default is **9600**.

Step 7	**stopbits** {**1** | **2**}
	Example:
	`switch(config-com1)# stopbits 2`
		Configures the stop bits. The default is **1**.

Step 8	**exit**
	Example:
	`switch(config-com1)# exit`
	`switch(config)#`
		Exits COM1 configuration mode.

Step 9	(Optional) **show line com1** **Example:** `switch(config)# show line com1`	Displays the COM1 port settings.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Virtual Terminals

This section describes how to configure virtual terminals on Cisco NX-OS devices.

Configuring the Inactive Session Timeout

You can configure a timeout for inactive virtual terminal sessions on a Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose

Step 3	- **exec-timeout** *minutes*
	- **absolute-timeout** *minutes*
	Example:
	`switch(config-line)# exec-timeout 30`
	Example:
	`switch(config-line)# absolute-timeout 30`
		Configures the inactive session timeout. The range is from 0 to 525600 minutes (8760 hours). A value of 0 minutes disables the timeout. The default value is 30.
		Sets a timeout interval on a virtual terminal (vty) line. The range is from 0 to 10000.
		The **absolute-timeout** command terminates the connection after the specified time period has elapsed, regardless of whether the connection is being used at the time of termination. You can specify an absolute-timeout value for each port. The user is given 20 seconds notice before the session is terminated. You can use this command along with the **logout-warning** command, which notifies the user of an impending logout.

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

Step 5	(Optional) **show running-config all | begin vty**
	Example:
	`switch(config)# show running-config all`
	` | begin vty`
		Displays the virtual terminal configuration.

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

Configuring the Session Limit

You can limit the number of virtual terminal sessions on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

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

Step 3	**session-limit** *sessions* **Example:** `switch(config-line)# session-limit 10`	Configures the maximum number of virtual sessions for the Cisco NX-OS device. The range is from 1 to 64. The default is 32.

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

Step 5	(Optional) **show running-config all | being vty** **Example:** `switch(config)# show running-config all | begin vty`	Displays the virtual terminal configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Modem Connections

You can connect a modem to either the COM1 port or the console port.

We recommend that you use the COM1 port to connect the modem.

Enabling a Modem Connection

You must enable the modem connection on the port before you can use the modem.

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 2</b></td><td>Enter one of the following commands:<table><tr><th>Command</th><th>Purpose</th></tr><tr><td><b>line com1</b></td><td>Enters COM1 configuration mode.</td></tr><tr><td><b>line console</b></td><td>Enters console configuration mode.</td></tr></table><b>Example:</b>
switch# line com1
switch(config-com1)#</td><td>Enters COM1 configuration mode or console configuration mode.</td></tr>
<tr><td><b>Step 3</b></td><td><b>modem in</b>
<b>Example:</b>
switch(config-com1)# modem in</td><td>Enables modem input on the COM1 or console port.</td></tr>
<tr><td><b>Step 4</b></td><td><b>exit</b>
<b>Example:</b>
switch(config-com1)# exit
switch(config)#</td><td>Exits COM1 or console configuration mode.</td></tr>
<tr><td><b>Step 5</b></td><td>(Optional) <b>show line</b>
<b>Example:</b>
switch(config)# show line</td><td>Displays the console and COM1 settings.</td></tr>
<tr><td><b>Step 6</b></td><td>(Optional) <b>copy running-config startup-config</b>
<b>Example:</b>
switch(config)# copy running-config
startup-config</td><td>Copies the running configuration to the startup configuration.</td></tr>
</table>

Downloading the Default Initialization String

The Cisco NX-OS software provides a default initialization string that you can download for connecting with the modem. The default initialization string is ATE0Q1&D2&C1S0=1\015.

Before you begin

Log in to the console port.

Procedure

	Command or Action	Purpose

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

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 2</b></td><td>Enter one of the following commands:
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td><b>line com1</b></td><td>Enters COM1 configuration mode.</td></tr>
<tr><td><b>line console</b></td><td>Enters console configuration mode.</td></tr>
</table>
<b>Example:</b>
<code>switch# line com1</code>
<code>switch(config-com1)#</code></td><td></td></tr>
<tr><td><b>Step 3</b></td><td><b>modem init-string default</b>
<b>Example:</b>
<code>switch(config-com1)# modem init-string default</code></td><td>Writes the default initialization string to the modem.</td></tr>
<tr><td><b>Step 4</b></td><td><b>exit</b>
<b>Example:</b>
<code>switch(config-com1)# exit</code>
<code>switch(config)#</code></td><td>Exits COM1 or console configuration mode.</td></tr>
<tr><td><b>Step 5</b></td><td>(Optional) <b>show line</b>
<b>Example:</b>
<code>switch(config)# show line</code></td><td>Displays the COM1 and console settings.</td></tr>
<tr><td><b>Step 6</b></td><td>(Optional) <b>copy running-config startup-config</b>
<b>Example:</b>
<code>switch(config)# copy running-config startup-config</code></td><td>Copies the running configuration to the startup configuration.</td></tr>
</table>

Configuring and Downloading a User-Specified Initialization String

You can configure and download your own initialization when the default initialization string is not compatible with your modem.

Before you begin

Log in to the console port.

Procedure

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

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td>switch# configure terminal
switch(config)#</td><td></td></tr>
<tr><td>Step 2</td><td>Enter one of the following commands:
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td><b>line com1</b></td><td>Enters COM1 configuration mode.</td></tr>
<tr><td><b>line console</b></td><td>Enters console configuration mode.</td></tr>
</table>
<b>Example:</b>
switch# line com1
switch(config-com1)#</td><td></td></tr>
<tr><td>Step 3</td><td><b>modem set-string user-input</b> <i>string</i>
<b>Example:</b>
switch(config-com1)# modem set-string
user-input ATE0Q1&D2&C1S0=3\015</td><td>Sets the user-specified initialization string for the COM1 or console port. The initialization string is alphanumeric and case sensitive, can contain special characters, and has a maximum of 100 characters.
<b>Note</b> You must first set the user-input string before initializing the string.</td></tr>
<tr><td>Step 4</td><td><b>modem init-string user-input</b>
<b>Example:</b>
switch(config-com1)# modem init-string
user-input</td><td>Writes the user-specified initialization string to the modem connected to the COM1 or console port.</td></tr>
<tr><td>Step 5</td><td><b>exit</b>
<b>Example:</b>
switch(config-com1)# exit
switch(config)#</td><td>Exits COM1 or console configuration mode.</td></tr>
<tr><td>Step 6</td><td>(Optional) <b>show line</b>
<b>Example:</b>
switch(config)# show line</td><td>Displays the COM1 and console settings.</td></tr>
<tr><td>Step 7</td><td>(Optional) <b>copy running-config startup-config</b>
<b>Example:</b>
switch(config)# copy running-config
startup-config</td><td>Copies the running configuration to the startup configuration.</td></tr>
</table>

Initializing a Modem for a Powered-Up Cisco NX-OS Device

If you connect a modem to a powered-up physical device, you must initialize the modem before you can use it.

Before you begin

After waiting until the Cisco NX-OS device has completed the boot sequence and the system image is running, connect the modem to either the COM1 port or the console port on the device.

Enable the modem connection on the port.

Procedure

	Command or Action	Purpose

Step 1	**modem connect line** {**com1** | **console**}
	Example:
	`switch# modem connect line com1`
		Initializes the modem connected to the device.

Related Topics

Enabling a Modem Connection, on page 122

Clearing Terminal Sessions

You can clear terminal sessions on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show users** **Example:** `switch# show users`	Displays the user sessions on the device.
Step 2	**clear line** *name* **Example:** `switch# clear line pts/0`	Clears a terminal session on a specific line. The line name is case sensitive.

Displaying Terminal and Session Information

To display terminal and session information, perform one of the following tasks:

Command	Purpose
show terminal	Displays terminal settings.
show line	Displays the COM1 and console ports settings.
show users	Displays virtual terminal sessions.
show running-config [**all**]	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.

For detailed information about the fields in the output from these commands, see the Cisco Nexus command reference guide for your device.

Default Settings for File System Parameters

This table lists the default settings for the file system parameters.

Table 19: Default File System Settings

Parameter	Default
Default filesystem	bootflash:

Additional References for Terminal Settings and Sessions

This section includes additional references for terminal settings and sessions on NX-OS devices.

Related Documents for Terminal Settings and Sessions

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*

CHAPTER 10

Basic Device Management

This chapter describes how to configure, manage, and verify the basic setting on your Cisco NX-OS device.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Default Settings for Basic Device Parameters

This table lists the default settings for basic device parameters.

Table 20: Default Basic Device Parameters

Parameters	Default
MOTD banner text	User Access Verification

Parameters	Default
Clock time zone	UTC

Information About Basic Device Management

This section provides information about basic device management.

Device Hostname

You can change the device hostname displayed in the command prompt from the default (switch) to another character string. When you give the device a unique hostname, you can easily identify the device from the command-line interface (CLI) prompt.

Message-of-the-Day Banner

The message-of-the-day (MOTD) banner displays before the user login prompt on the device. This message can contain any information that you want to display for users of the device.

EXEC Banner

Starting with the Cisco NX-OS Release 7.3(0)D1(1), the EXEC banner is displayed after a user logs in to a switch. This banner can be used to post reminders to your network administrators.

Device Clock

If you do not synchronize your device with a valid outside timing mechanism, such as an NTP clock source, you can manually set the clock time when your device boots.

Clock Manager

The Cisco Nexus chassis may contain clocks of different types that may need to be synchronized. These clocks are a part of various components (such as the supervisor, LC processors, or line cards) and each may be using a different protocol.

The clock manager provides a way to synchronize these different clocks.

Time Zone and Summer Time (Daylight Saving Time)

You can configure the time zone and summer time (daylight saving time) setting for your device. These values offset the clock time from Coordinated Universal Time (UTC). UTC is International Atomic Time (TAI) with leap seconds added periodically to compensate for the Earth's slowing rotation. UTC was formerly called Greenwich Mean Time (GMT).

User Sessions

You can display the active user session on your device. You can also send messages to the user sessions. For more information about managing user sessions and accounts, see the Cisco Nexus security configuration guide for your device.

Changing the Device Hostname

You can change the device hostname displayed in the command prompt from the default (switch) to another character string.

Procedure

	Command or Action	Purpose

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

Step 2	{**hostname** | **switchname**} *name*
	Example:
	Using the **hostname** command:
	`switch(config)# hostname Engineering1`
	`Engineering1(config)#`
	Using the **switchname** command:
	`Engineering1(config)# switchname`
	`Engineering2`
	`Engineering2(config)#`
		Note The **switchname** command performs the same function as the **hostname** command.

Step 3	**exit**
	Example:
	`Engineering2(config)# exit`
	`Engineering2#`
		Exits global configuration mode.

Step 4	(Optional) **copy running-config startup-config**
	Example:
	`Engineering2# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring the MOTD Banner

You can configure the MOTD to display before the login prompt on the terminal when a user logs in. The MOTD banner has the following characteristics:

- Maximum of 254 characters per line

- Maximum of 40 lines

Procedure

	Command or Action	Purpose

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

Step 2	**banner motd** *delimiting-character message delimiting-character*
	Example:
	`switch(config)# banner motd #Welcome to the Switch#`
	`switch(config)#`
		Configures the MOTD banner. Do not use the *delimiting-character* in the *message* text.
		Note Do not use " or % as a delimiting character.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show banner motd** **Example:** `switch# show banner motd`	Displays the configured MOTD banner.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the EXEC Banner

You can configure the EXEC banner to display a message when a user logs in to a device. The EXEC banner has the following characteristics:

- Maximum of 254 characters per line including the delimiting characters
- Maximum of 40 lines

Procedure

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

	Command or Action	Purpose

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

Step 2	**banner exec** *delimiting-character message delimiting-character*
	Example:
	`switch(config)# banner exec #Welcome to the Test#`
	`switch(config)#`
		Configures the EXEC banner. Do not use the *delimiting-character* in the *message* text.

Step 3	(Optional) **no banner exec** **Example:** `switch(config)# no banner exec`	Resets the value of EXEC banner to the default value. **Note** The default value of the EXEC banner is blank.

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

Step 5	(Optional) **show banner exec** **Example:** `switch# show banner exec`	Displays the configured EXEC banner.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the EXEC Banner

This example shows how to configure the EXEC banner.

```
# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# banner exec #Unauthorized access to this device is prohibited!#
switch(config)# exit
switch# show banner exec
Unauthorized access to this device is prohibited!
```

Configuring the Time Zone

You can configure the time zone to offset the device clock time from UTC.

Procedure

	Command or Action	Purpose

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

Step 2	**clock timezone** *zone-name offset-hours offset-minutes* **Example:** `switch(config)# clock timezone EST -5 0`	Configures the time zone. The *zone-name* argument is a 3-character string for the time zone acronym (for example, PST or EST). The *offset-hours* argument is the offset from the UTC and the range is from –23 to 23 hours. The range for the *offset-minutes* argument is from 0 to 59 minutes.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show clock** **Example:** `switch# show clock`	Displays the time and time zone.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Summer Time (Daylight Saving Time)

You can configure when summer time, or daylight saving time, is in effect for the device and the offset in minutes.

Procedure

	Command or Action	Purpose

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

Step 2	**clock summer-time** *zone-name start-week start-day start-month start-time end-week end-day end-month end-time offset-minutes*	Configures summer time or daylight saving time.

	Command or Action	Purpose
	Example: `switch(config)# clock summer-time PDT 1 Sunday March 02:00 1 Sunday November 02:00 60`	The *zone-name* argument is a three character string for the time zone acronym (for example, PST and EST). The values for the *start-day* and *end-day* arguments are **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, and **Sunday**. The values for the *start-month* and *end-month* arguments are **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**. The value for the *start-time* and *end-time* arguments are in the format *hh***:***mm*. The range for the *offset-minutes* argument is from 0 to 1440 minutes.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show clock detail** **Example:** `switch(config)# show clock detail`	Displays the configured MOTD banner.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Manually Setting the Device Clock

You can set the clock manually if your device cannot access a remote time source.

Before you begin

Configure the time zone.

Procedure

	Command or Action	Purpose

Step 1	**clock set** *time day month year*
	Example:
	`switch# clock set 15:00:00 30 May 2008`
	`Fri May 30 15:14:00 PDT 2008`
		Configures the device clock.
		The format for the *time* argument is *hh***:***mm***:***ss*.
		The range for the *day* argument is from 1 to 31.

	Command or Action	Purpose
		The values for the *month* argument are **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**. The range for the *year* argument is from 2000 to 2030.
Step 2	(Optional) **show clock** **Example:** `switch(config)# show clock`	Displays the current clock value.

Related Topics

Configuring the Time Zone, on page 133

Setting the Clock Manager

You can configure the clock manager to synchronize all the clocks of the components in the Cisco Nexus chassis.

Procedure

	Command or Action	Purpose
Step 1	**clock protocol** *protocol* **vdc** *vdc-num* **Example:** `# clock protocol ptp vdc 2`	Configures the clock manager. The values for the *protocol* argument are **ptp**, **ntp**, and **none**. The following describes the values: • **ptp**—Synchronizes clocks with Precision Time Protocol (PTP) as described by IEEE 1588. • **ntp**— Synchronizes clocks with Network Time Protocol (NTP). • **none**—Use **clock set** to set supervisor clocks. **Note** When **none** is used, the clock in the specified VDC must be configured.

	Command or Action	Purpose
		Note Once the protocol is configured, the clock in the specified VDC must use that protocol. For example, if the **clock protocol ptp vdc 2** command is entered, then PTP should be configured in VDC 2. The range for the *vdc* argument is 1 to 8.
Step 2	(Optional) **show run clock_manager** **Example:** `#show run clock_manager`	Displays the configuration of the clock manager.

Managing Users

You can display information about users logged into the device and send messages to those users.

Displaying Information about the User Sessions

You can display information about the user session on the device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays the user sessions.

Sending a Message to Users

You can send a message to active users currently using the device CLI.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show users** **Example:** `switch# show users`	Displays the active user sessions.

	Command or Action	Purpose
Step 2	**send** [**session** *line*] *message-text* **Example:** `switch# send Reloading the device is 10 minutes!`	Sends a message to all active users or to a specific user. The message can be up to 80 alphanumeric characters and is case sensitive.

Verifying the Device Configuration

To verify the configuration after bootstrapping the device using POAP, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.

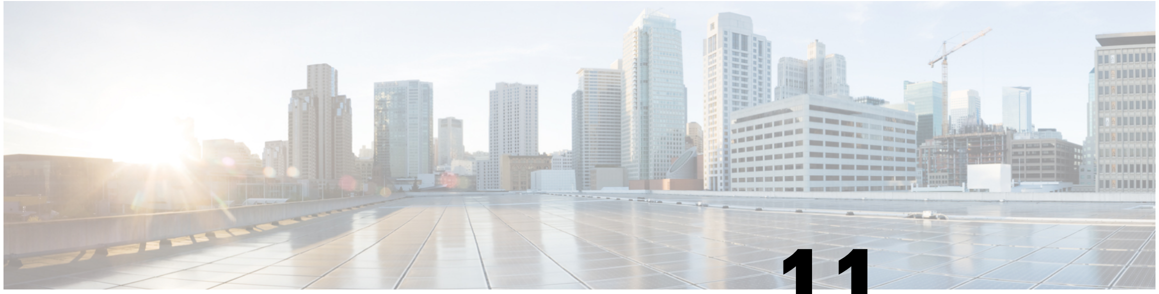
For detailed information about the fields in the output from these commands, see the Cisco Nexus command reference for your device.

Additional References for Basic Device Management

You can find additional information related to basic device management.

Related Documents for Basic Device Management

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*



CHAPTER 11

Using the Device File Systems, Directories, and Files

This chapter describes how to use your device file systems, directories, and files.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Information About Device File Systems, Directories, Files, and External Storage Devices

This section describes the file systems, directories, files, and support provided to the external storage devices on Cisco NX-OS devices.

File Systems

This topic provides information about the file system components supported on a Cisco MDS device. (The syntax for specifying a local file system is *filesystem***:**[//*modules*/].)

Note The default *filesystem* parameter is bootflash:.

This table describes the file system components that you can use on a Cisco MDS device.

Table 21: File System Components

File System Name	Module	Description
bootflash	sup-active sup-local	Internal CompactFlash memory located on an active supervisor module. Used for storing image files, configuration files, and other miscellaneous files. The initial default directory is bootflash.
	sup-standby sup-remote	Internal CompactFlash memory located on a standby supervisor module. Used for storing image files, configuration files, and other miscellaneous files.
volatile	—	Volatile random-access memory (VRAM) located on a supervisor module. Used for temporary or pending changes.
log	—	Memory on an active supervisor module. Used for storing file statistics logs.
system	—	Memory on a supervisor module. Used for storing the running configuration file.
debug	—	Memory on a supervisor module. Used for storing the debug logs.

Directories

You can create directories on bootflash: and external flash memory (slot0:, usb1:, and usb2:). You can create, store, and access files from directories.

Files

You can create and access files from bootflash:, volatile:, slot0:, usb1:, and usb2: file systems. You can only access files from the system: file system. Use the debug: file system to store the debug log files specified using the **debug logfile** command.

You can download files, such as system image files, from remote servers using FTP, Secure Copy Protocol (SCP), Secure File Transfer Protocol (SFTP), and TFTP. You can also copy files from an external server to your device because your device can act as an SCP server.

Working with Directories

This section describes how to work with directories on a Cisco NX-OS device.

Identifying the Current Directory

You can display the directory name of your current directory.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	Displays the name of your current directory.

Changing the Current Directory

You can change the current directory for file system operations. The initial default directory is bootflash:.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.

Step 2	**cd** {*directory* | *filesystem***:**[//*module*/][*directory*]}
	Example:
	`switch# cd slot0:`
		Changes to a new current directory. The file system, module, and directory names are case sensitive.

Creating a Directory

You can create directories in the bootflash: and flash device file systems.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.

Step 2	(Optional) **cd** {*directory* | *filesystem*:[//*module*/][*directory*]}
	Example:
	`switch# cd slot0:`
		Changes to a new current directory. The file system, module, and directory names are case sensitive.

Step 3	**mkdir** [*filesystem*:[//*module*/]]*directory* **Example:** `switch# mkdir test`	Creates a new directory. The *filesystem* argument is case sensitive. The *directory* argument is alphanumeric, case sensitive, and has a maximum of 64 characters.

Displaying Directory Contents

You can display the contents of a directory.

Procedure

	Command or Action	Purpose
Step 1	**dir** [*directory* | *filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the directory contents. The default is the current working directory. The file system and directory names are case sensitive.

Deleting a Directory

You can remove directories from the file systems on your device.

Before you begin

Ensure that the directory is empty before you try to delete it.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.

	Command or Action	Purpose
Step 2	(Optional) **dir** [*filesystem* **:**[//*module*/][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the contents of the current directory. The file system, module, and directory names are case sensitive. If the directory is not empty, you must delete all the files before you can delete the directory.
Step 3	**rmdir** [*filesystem* **:**[//*module*/]]*directory* **Example:** `switch# rmdir test`	Deletes a directory. The file system and directory name are case sensitive.

Accessing the Directories on a Standby Supervisor Module

You can access all the file systems on a standby supervisor module (remote) from a session on an active supervisor module. This feature is useful when copying files to the active supervisor module that requires similar files to exist, as in the standby supervisor module.

To access the file systems on the standby supervisor module from a session on the active supervisor module, specify the standby supervisor module in the path to the file using either the *filesystem*:**//sup-remote/** command, or the *filesystem*:**//sup-standby/** command.

Working with Files

This section describes how to work with files on a Cisco NX-OS device.

Moving Files

You can move a file from one directory to another directory.

Caution If a file with the same name already exists in the destination directory, that file is overwritten by the moved file.

You can use the **move** command to rename a file by moving the file within the same directory.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **dir** [*filesystem*:[//*module*/][*directory*]] **Example:**	Displays the contents of the current directory. The file system and directory name are case sensitive.

	Command or Action	Purpose
	`switch# dir bootflash`	

Step 3	**move** [*filesystem***:**[*//module/*][*directory /*] | *directory/*]*source-filename* {{*filesystem***:**[*//module/*][*directory /*] | *directory/*}[*target-filename*] | *target-filename*}
	Example:
	`switch# move test old_tests/test1`
		Moves a file.
		The file system, module, and directory names are case sensitive.
		The *target-filename* argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the *target-filename* argument is not specified, the filename defaults to the *source-filename* argument value.

Copying Files

You can make copies of files, either within the same directory or on another directory.

Note Use the **dir** command to ensure that enough space is available in the target file system. If enough space is not available, use the **delete** command to remove unneeded files.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **dir** [*filesystem***:**[*//module/*][*directory*]] **Example:** `switch# dir bootflash`	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 3	**copy** [*filesystem***:**[*//module/*][*directory/*] | *directory/*]*source-filename* | {*filesystem***:**[*//module/*][*directory/*]] | *directory/*}[*target-filename*] **Example:** `switch# copy test old_tests/test1`	Copies a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the *target-filename* argument is not specified, the filename defaults to the *source-filename* argument value. The copy command supports ftp, scp, sftp, tftp and http protocols.

Deleting Files

You can delete a file from a directory.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** [*filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash`	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	**delete** {*filesystem*:[//*module*/][*directory*/] | *directory*/}*filename* **Example:** `switch# delete test old_tests/test1`	Deletes a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is case sensitive. **Caution** If you specify a directory, the **delete** command deletes the entire directory and all its contents.

Displaying File Contents

You can display the contents of a file.

Procedure

	Command or Action	Purpose
Step 1	**show file** [*filesystem*:[//*module*/]][*directory*/]*filename* **Example:** `switch# show file bootflash:test-results`	Displays the file contents.

Displaying File Checksums

You can display checksums to check the file integrity.

Procedure

	Command or Action	Purpose

Step 1	**show file** [*filesystem*:[//*module*/]][*directory*/]*filename* {**cksum** | **md5sum**}
	Example:
	`switch# show file bootflash:trunks2.cfg cksum`
		Displays the checksum or MD5 checksum of the file.

Compressing and Uncompressing Files

You can compress and uncompress files on your Cisco NX-OS device using Lempel-Ziv 1977 (LZ77) coding.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** [*filesystem***:**[*//module/*]*directory*]] **Example:** `switch# dir bootflash:`	Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	**gzip** [*filesystem***:**[*//module/*][*directory/*] | *directory/*]*filename* **Example:** `switch# gzip show_tech`	Compresses a file. After the file is compressed, it has a .gz suffix.
Step 3	**gunzip** [*filesystem***:**[*//module/*][*directory/*] | *directory/*]*filename* **.gz** **Example:** `switch# gunzip show_tech.gz`	Uncompresses a file. The file to uncompress must have the .gz suffix. After the file is uncompressed, it does not have the .gz suffix.

Displaying the Last Lines in a File

You can display the last lines of a file.

Procedure

	Command or Action	Purpose
Step 1	**tail** [*filesystem***:**[*//module/*]][*directory/*]*filename* [*lines*] **Example:** `switch# tail ospf-gr.conf`	Displays the last lines of a file. The default number of lines is 10. The range is from 0 to 80 lines.

Redirecting show Command Output to a File

You can redirect **show** command output to a file on bootflash:, slot0:, volatile:, or on a remote server.

Procedure

	Command or Action	Purpose
Step 1	*show-command* > [*filesystem***:**[*//module/*][*directory*] | [directory /]]*filename* **Example:**	Redirects the output from a **show** command to a file.

	Command or Action	Purpose
	`switch# show tech-support > bootflash:techinfo`	

Finding Files

You can find the files in the current working directory and its subdirectories that have names that begin with a specific character string.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.

Step 2	(Optional) **cd** {*filesystem*:[//*module*/][*directory*] | *directory*}
	Example:
	`switch# cd bootflash:test_scripts`
		Changes the default directory.

Step 3	**find** *filename-prefix* **Example:** `switch# find bgp_script`	Finds all filenames in the default directory and in its subdirectories beginning with the filename prefix. The filename prefix is case sensitive.

Working with Archive Files

The Cisco NX-OS software supports archive files. Besides creating an archive file, you can append files to, extract files from, and list the files in an archive file.

Creating an Archive Files

You can create an archive file and add files to it. You can specify the following compression types:

- bzip2
- gzip
- Uncompressed

The default is gzip.

Procedure

	Command or Action	Purpose
Step 1	**tar create** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**bz2-compress**] [**gz-compress**] [**remove**] [**uncompressed**] [**verbose**] *filename-list*	Creates an archive file and adds files to it. The filename is alphanumeric, not case sensitive, and has a maximum length of 240 characters. The **absolute** keyword specifies that the leading backslash characters (\) should not be removed from the names of the files added to the archive file. By default, the leading backslash characters are removed. The **bz2-compress**, **gz-compress**, and **uncompressed** keywords determine the compression utility used when files are added, or later appended, to the archive and the decompression utility to use when extracting the files. If you do not specify an extension for the archive file, the defaults are as follows: • For **bz2-compress**, the extension is .tar.bz2. • For **gz-compress**, the extension is .tar.gz. • For **uncompressed**, the extension is .tar. The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the file system after adding them to the archive. By default, the files are not deleted. The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

Example

This example shows how to create a gzip compressed archive file:

```
switch# tar create bootflash:config-archive gz-compress bootflash:config-file
```

Appending Files to an Archive File

You can append files to an existing archive file on your Cisco NX-OS device.

Before you begin

You have created an archive file on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**tar append** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**remove**] [**verbose**] *filename-list*	Adds files to an existing archive file. The archive filename is not case sensitive. The **absolute** keyword specifies that the leading backslash characters (\) should not be removed from the names of the files added to the archive file. By default, the leading backslash characters are removed. The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the filesystem after adding them to the archive. By default, the files are not deleted. The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

Example

This example shows how to append a file to an existing archive file:

```
switch# tar append bootflash:config-archive.tar.gz bootflash:new-config
```

Extracting Files from an Archive File

You can extract files to an existing archive file on your Cisco NX-OS device.

Before you begin

You have created an archive file on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**tar extract** {**bootflash:** | **volatile:**}*archive-filename* [**keep-old**] [**screen**] [**to** {**bootflash:** | **volatile:**}[*/directory-name*]] [**verbose**]	Extracts files from an existing archive file. The archive filename is not case sensitive. The **keep-old** keyword indicates that the Cisco NX-OS software should not overwrite files with the same name as the files being extracted. The **screen** keyword specifies that the Cisco NX-OS software should display the contents of the extracted files to the terminal screen. The **to** keyword specifies the target file system. You can include a directory name. The directory

	Command or Action	Purpose
		name is alphanumeric, case sensitive, and has a maximum length of 240 characters. The **verbose** keyword specifies that the Cisco NX-OS software should display the names of the files as they are extracted.

Example

This example shows how to extract files from an existing archive file:

```
switch# tar extract bootflash:config-archive.tar.gz
```

Displaying the Filenames in an Archive File

Note The archive filename is not case sensitive.

To display the file names in an archive file, run the following command:

tar list {**bootflash:** | **volatile:**}*archive-filename*

Example:

```
switch# tar list bootflash:config-archive.tar.gz
config-file
new-config
```

Examples of Using a File System

This section includes examples of using a file system on a Cisco NX-OS device.

Accessing Directories on a Standby Supervisor Module

This example shows how to list the files on a standby supervisor module:

```
switch# dir bootflash://sup-remote
   12198912     Aug 27 16:29:18 2003  m9500-sf1ek9-kickstart-mzg.1.3.0.39a.bin
    1864931     Apr 29 12:41:59 2003  dplug2
      12288     Apr 18 20:23:11 2003  lost+found/
   12097024     Nov 21 16:34:18 2003  m9500-sf1ek9-kickstart-mz.1.3.1.1.bin
   41574014     Nov 21 16:34:47 2003  m9500-sf1ek9-mz.1.3.1.1.bin

Usage for bootflash://sup-remote
   67747169 bytes used
  116812447 bytes free
  184559616 bytes total
```

This example shows how to delete a file on a standby supervisor module:

```
switch# delete bootflash://sup-remote/aOldConfig.txt
```

Moving Files

This example shows how to move a file on an external flash device:

```
switch# move slot0:samplefile slot0:mystorage/samplefile
```

This example shows how to move a file in the default file system:

```
switch# move samplefile mystorage/samplefile
```

Copying Files

This example shows how to copy a file called samplefile from the root directory of the slot0: file system to the mystorage directory:

```
switch# copy slot0:samplefile slot0:mystorage/samplefile
```

This example shows how to copy a file from the current directory:

```
switch# copy samplefile mystorage/samplefile
```

This example shows how to copy a file from an active supervisor module bootflash to a standby supervisor module bootflash:

```
switch# copy bootflash:system_image bootflash://sup-2/system_image
```

Note You can also use the **copy** command to upload and download files from the slot0: or bootflash: file system to or from an FTP, TFTP, SFTP, or SCP server.

Deleting a Directory

You can remove directories from the file systems on your device.

Before you begin

Ensure that the directory is empty before you try to delete it.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **pwd** **Example:** `switch# pwd`	Displays the name of your current default directory.
Step 2	(Optional) **dir** [*filesystem* **:**[//*module*/]][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the contents of the current directory. The file system, module, and directory names are case sensitive. If the directory is not empty, you must delete all the files before you can delete the directory.
Step 3	**rmdir** [*filesystem* **:**[//*module*/]]*directory* **Example:** `switch# rmdir test`	Deletes a directory. The file system and directory name are case sensitive.

Displaying File Contents

This example shows how to display the contents of a file on an external flash device:

```
switch# show file slot0:test
configure terminal
interface ethernet 1/1
no shutdown
end
show interface ethernet 1/1
```

This example shows how to display the contents of a file that resides in the current directory:

```
switch# show file myfile
```

Displaying File Checksums

This example shows how to display the checksum of a file:

```
switch# show file bootflash:trunks2.cfg cksum
583547619
```

This example shows how to display the MD5 checksum of a file:

```
switch# show file bootflash:trunks2.cfg md5sum
3b94707198aabefcf46459de10c9281c
```

Compressing and Uncompressing Files

This example shows how to compress a file:

```
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
...
switch# gzip volatile:Samplefile
switch# dir
     266069     Jul 04 00:51:03 2003 Samplefile.gz
...
```

This example shows how to uncompress a compressed file:

```
switch# dir
     266069     Jul 04 00:51:03 2003 Samplefile.gz
...
switch# gunzip samplefile
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
...
```

Redirecting show Command Output

This example shows how to direct the output to a file on the bootflash: file system:

```
switch# show interface > bootflash:switch1-intf.cfg
```

This example shows how to direct the output to a file on external flash memory:

```
switch# show interface > slot0:switch-intf.cfg
```

This example shows how to direct the output to a file on a TFTP server:

```
switch# show interface > tftp://10.10.1.1/home/configs/switch-intf.cfg
Preparing to copy...done
```

This example shows how to direct the output of the **show tech-support** command to a file:

```
switch# show tech-support > Samplefile
Building Configuration ...
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
Usage for volatile://
    1527808 bytes used
   19443712 bytes free
   20971520 bytes total
```

Finding Files

This example shows how to find a file in the current default directory:

```
switch# find smm_shm.cfg
/usr/bin/find: ./lost+found: Permission denied
```

```
./smm_shm.cfg
./newer-fs/isan/etc/routing-sw/smm_shm.cfg
./newer-fs/isan/etc/smm_shm.cfg
```

Default Settings for File System Parameters

This table lists the default settings for the file system parameters.

Table 22: Default File System Settings

Parameter	Default
Default filesystem	bootflash:

Additional References for File Systems

This section includes additional information related to the file systems.

Related Documents for File Systems

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*

CHAPTER 12

Working with Configuration Files

This chapter describes how to work with your device configuration files.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Information About Configuration Files

Configuration files contain the Cisco NX-OS software commands used to configure the features on a Cisco NX-OS device. Commands are parsed (translated and executed) by the Cisco NX-OS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

To change the startup configuration file, you can either save the running-configuration file to the startup configuration using the **copy running-config startup-config** command or copy a configuration file from a file server to the startup configuration.

Types of Configuration Files

The Cisco NX-OS software has two types of configuration files, running configuration and startup configuration. The device uses the startup configuration (startup-config) during device startup to configure the software features. The running configuration (running-config) contains the current changes that you make to the startup-configuration file. The two configuration files can be different. You might want to change the device configuration for a short time period rather than permanently. In this case, you would change the running configuration by using commands in global configuration mode but not save the changes to the startup configuration.

To change the running configuration, use the **configure terminal** command to enter global configuration mode. As you use the Cisco NX-OS configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup-configuration file, you can either save the running configuration file to the startup configuration or download a configuration file from a file server to the startup configuration.

Related Topics

Saving the Running Configuration to the Startup Configuration, on page 156
Downloading the Startup Configuration From a Remote Server, on page 158

Managing Configuration Files

This section describes how to manage configuration files.

Saving the Running Configuration to the Startup Configuration

You can save the running configuration to the startup configuration to save your changes for the next time you that reload the device.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the running configuration.
Step 2	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Copying a Configuration File to a Remote Server

You can copy a configuration file stored in the internal memory to a remote server as a backup or to use for configuring other Cisco NX-OS devices.

Procedure

	Command or Action	Purpose
Step 1	**copy running-config** *scheme***://***server***/**[*url* **/**]*filename* **Example:**	Copies the running-configuration file to a remote server.

	Command or Action	Purpose
	`switch# copy running-config tftp://10.10.1.1/sw1-run-config.bak`	For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.
Step 2	**copy startup-config** *scheme***://***server***/**[*url* **/**]*filename* **Example:** `switch# copy startup-config tftp://10.10.1.1/sw1-start-config.bak`	Copies the startup-configuration file to a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.

Example

This example shows how to copy the configuration file to a remote server:

```
switch# copy running-config
tftp://10.10.1.1/sw1-run-config.bak
switch# copy startup-config
tftp://10.10.1.1/sw1-start-config.bak
```

Downloading the Running Configuration From a Remote Server

You can configure your Cisco NX-OS device by using configuration files that you created on another Cisco NX-OS device and uploaded to a remote server. You then download the file from the remote server to your device using TFTP, FTP, Secure Copy (SCP), or Secure Shell FTP (SFTP) to the running configuration.

Before you begin

Ensure that the configuration file that you want to download is in the correct directory on the remote server.

Ensure that the permissions on the file are set correctly. Permissions on the file should be set to world-read.

Ensure that your Cisco NX-OS device has a route to the remote server. The Cisco NX-OS device and the remote server must be in the same subnetwork if you do not have a router or a default gateway to route traffic between subnets.

Check connectivity to the remote server using the **ping** or **ping6** command.

Procedure

	Command or Action	Purpose

Step 1	**copy** *scheme*://*server*/[*url*/]*filename* **running-config**
	Example:
	`switch# copy tftp://10.10.1.1/my-config`
	`running-config`
		Downloads the running-configuration file from a remote server.
		For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.
		The *server*, *url*, and *filename* arguments are case sensitive.

Step 2	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the running configuration.

Step 3	(Optional) **copy running-config startup-config**
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Step 4	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the startup configuration.

Downloading the Startup Configuration From a Remote Server

You can configure your Cisco NX-OS device by using configuration files that you created on another Cisco NX-OS device and uploaded to a remote server. You then download the file from the remote server to your device using TFTP, FTP, Secure Copy (SCP), or Secure Shell FTP (SFTP) to the startup configuration.

Caution This procedure disrupts all traffic on the Cisco NX-OS device.

Before you begin

Log in to a session on the console port.

Ensure that the configuration file that you want to download is in the correct directory on the remote server.

Ensure that the permissions on the file are set correctly. Permissions on the file should be set to world-read.

Ensure that your Cisco NX-OS device has a route to the remote server. The Cisco NX-OS device and the remote server must be in the same subnetwork if you do not have a router or a default gateway to route traffic between subnets.

Check connectivity to the remote server using the **ping** or **ping6** command.

Procedure

	Command or Action	Purpose
Step 1	**write erase** **Example:** `switch# write erase`	Erases the startup configuration file.

Step 2	**reload**
	Example:
	`switch# reload`
	`This command will reboot the system.`
	`(y/n)?  [n] y`
	`...`
	`Enter the password for "admin":`
	`<password>`
	`Confirm the password for "admin":`
	`<password>`
	`...`
	`Would you like to enter the basic`
	`configuration`
	`dialog (yes/no): n`
	`switch#`
		Reloads the Cisco NX-OS device.
		Note Do not use the setup utility to configure the device.
		Note By default, the **reload** command reloads the device from a binary version of the startup configuration.
		Beginning with Cisco NX-OS 6.2(2), you can use the **reload ASCII** command to copy an ascii version of the configuration to the start up configuration when reloading the device.

Step 3	**copy** *scheme***://***server***/**[*url* **/**]*filename* **running-config**
	Example:
	`switch# copy tftp://10.10.1.1/my-config`
	`running-config`
		Downloads the running configuration file from a remote server.
		For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.
		The *server*, *url*, and *filename* arguments are case sensitive.

Step 4	**copy running-config startup-config**
	Example:
	`switch# copy running-config`
	`startup-config`
		Saves the running configuration file to the startup configuration file.

Step 5	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the running configuration.

Copying Configuration Files to an External Flash Memory Device

You can copy configuration files to an external flash memory device as a backup for later use.

Before you begin

Insert the external Flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*] **Example:** `switch# dir slot0:`	Displays the files on the external flash memory device.
Step 2	**copy running-config** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **Example:** `switch# copy running-config slot0:dsn-running-config.cfg`	Copies the running configuration to an external flash memory device. The *filename* argument is case sensitive.
Step 3	**copy startup-config** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **Example:** `switch# copy startup-config slot0:dsn-startup-config.cfg`	Copies the startup configuration to an external flash memory device. The *filename* argument is case sensitive.

Copying the Running Configuration from an External Flash Memory Device

You can configure your Cisco NX-OS device by copying configuration files created on another Cisco NX-OS device and saved to an external flash memory device.

Before you begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*] **Example:** `switch# dir slot0:`	Displays the files on the external flash memory device.
Step 2	**copy** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **running-config** **Example:** `switch# copy slot0:dsn-config.cfg running-config`	Copies the running configuration from an external flash memory device. The *filename* argument is case sensitive.
Step 3	(Optional) **show running-config** **Example:** `switch# show running-config`	Displays the running configuration.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 5	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the startup configuration.

Copying the Startup Configuration from an External Flash Memory Device

You can recover the startup configuration on your Cisco NX-OS device by downloading a new startup configuration file saved on an external flash memory device.

Before you begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*] **Example:** `switch# dir slot0:`	Displays the files on the external flash memory device.
Step 2	**copy** {**slot0:** | **usb1:** | **usb2:**}[*directory* /]*filename* **startup-config** **Example:** `switch# copy slot0:dsn-config.cfg startup-config`	Copies the startup configuration from an external flash memory device. The *filename* argument is case sensitive.
Step 3	(Optional) **show startup-config** **Example:** `switch# show startup-config`	Displays the startup configuration.

Copying Configuration Files to an Internal File System

You can copy configuration files to the internal memory as a backup for later use.

Procedure

	Command or Action	Purpose
Step 1	**copy running-config** [*filesystem***:**][*directory/*] | [*directory/*]*filename* **Example:** `switch# copy running-config bootflash:sw1-run-config.bak`	Copies the running-configuration file to internal memory. The *filesystem*, *directory*, and *filename* arguments are case sensitive.
Step 2	**copy startup-config** [*filesystem***:**][*directory/*] | [*directory/*]*filename* **Example:** `switch# copy startup-config bootflash:sw1-start-config.bak`	Copies the startup-configuration file to internal memory. The *filesystem*, *directory*, and *filename* arguments are case sensitive.

Related Topics

Copying Files, on page 144

Rolling Back to a Previous Configuration

Problems, such as memory corruption, can occur that make it necessary for you to recover your configuration from a backed up version.

Note Each time that you enter a **copy running-config startup-config** command, a binary file is created and the ASCII file is updated. A valid binary configuration file reduces the overall boot time significantly. A binary file cannot be uploaded, but its contents can be used to overwrite the existing startup configuration. The **write erase** command clears the binary file.

Procedure

	Command or Action	Purpose
Step 1	**write erase** **Example:** `switch# write erase`	Clears the current configuration of the switch.
Step 2	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run.

	Command or Action	Purpose
		Note By default, the **reload** command reloads the device from a binary version of the startup configuration. Beginning with Cisco NX-OS 6.2(2), you can use the **reload ascii** command to copy an ASCII version of the configuration to the start up configuration when reloading the device.
Step 3	**copy** *configuration_file* **running-configuration** **Example:** `switch# copy bootflash:start-config.bak running-configuration`	Copies a previously saved configuration file to the running configuration. **Note** The *configuration_file* filename argument is case sensitive.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the start-up configuration.

Removing the Configuration for a Missing Module

When you remove an I/O module from the chassis, you can also remove the configuration for that module from the running configuration.

Note You can only remove the configuration for an empty slot in the chassis.

Before you begin

Remove the I/O module from the chassis.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show hardware** **Example:** `switch# show hardware`	Displays the installed hardware for the device.
Step 2	**purge module** *slot* **running-config** **Example:** `switch# purge module 3 running-config`	Removes the configuration for a missing module from the running configuration.

	Command or Action	Purpose
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Erasing a Configuration

You can erase the configuration on your device to return to the factory defaults.

You can erase the following configuration files saved in the persistent memory on the device:

- Startup
- Boot
- Debug

The **write erase** command erases the entire startup configuration, except for the following:

- Boot variable definitions
- The IPv4 configuration on the mgmt0 interface, including the following:
    - Address
    - Subnet mask

To remove the boot variable definitions follow step-1 and step-2.

To remove the boot variables, running configuration, and the IP configuration on the management interface follow step-3 to step-5.

Procedure

	Command or Action	Purpose
Step 1	**write erase boot** **Example:** `switch# write erase boot`	Erases the boot variable definitions.
Step 2	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run. By default, the reload command reloads the device from a binary version of the startup configuration.
Step 3	**write erase** **Example:** `switch# write erase`	Erases the boot variable definitions.

	Command or Action	Purpose
Step 4	**write erase boot** **Example:** `switch# write erase boot`	Erases the boot variable definitions and the IPv4 configuration on the management interface.
Step 5	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run. By default, the reload command reloads the device from a binary version of the startup configuration.

Clearing Inactive Configurations

You can clear inactive Quality of Service (QoS) and/or access control list (ACL) configurations.

Procedure

	Command or Action	Purpose
Step 1	(Optional) **show running-config** *type* **inactive-if-config** **Example:** `# show running-config ipqos inactive-if-config`	Displays any inactive ACL or QoS configurations. The values for the *type* argument are **aclmgr** and **ipqos**. • **aclmgr**— Displays any inactive configurations for aclmgr. • **ipqos**—Displays any inactive configurations for qosmgr.

Step 2	**clear inactive-config** *policy*
	Example:
	`# clear inactive-config qos`
	`clear qos inactive config`
	`Inactive if config for QoS manager is saved at/bootflash/qos_inactive_if_config.cfg`
	
	`for vdc default & for other than default vdc: /bootflash/vdc_x/qos_inactive_if_config.cfg (where x is vdc number)`
	`you can see the log file @ show inactive-if-config log`
		Clears inactive configurations.
		The values for the *policy* argument are **qos** and **acl**.
		The following describes the values:
		• **qos**—Clears inactive QoS configurations.
		• **acl**— Clears inactive ACL configurations.
		• **acl qos**—Clears inactive ACL configurations and inactive QoS configurations.

Step 3	(Optional) **show inactive-if-config log** **Example:** `# show inactive-if-config log`	Displays the commands that were used to clear the inactive configurations.

Verifying the Device Configuration

To verify the configuration after bootstrapping the device using POAP, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.

For detailed information about the fields in the output from these commands, see the Cisco Nexus command reference for your device.

Examples of Working with Configuration Files

This section includes examples of working with configuration files.

Copying Configuration Files

This example shows how to copy a running configuration to the bootflash: file system:

Backing Up Configuration Files

This example shows how to back up the startup configuration to the bootflash: file system (ASCII file):

```
switch# copy startup-config bootflash:my-config
```

This example shows how to back up the startup configuration to the TFTP server (ASCII file):

```
switch# copy startup-config tftp://172.16.10.100/my-config
```

This example shows how to back up the running configuration to the bootflash: file system (ASCII file):

```
switch# copy running-config bootflash:my-config
```

Rolling Back to a Previous Configuration

To roll back your configuration to a snapshot copy of a previously saved configuration, you need to perform the following steps:

1. Clear the current running image with the **write erase** command.
2. Restart the device with the **reload** command.

Note By default, the **reload** command reloads the device from a binary version of the startup configuration.

Beginning with Cisco NX-OS 6.2(2), you can use the **reload ascii** command to copy an ASCII version of the configuration to the start up configuration when reloading the device.

3. Copy the previously saved configuration file to the running configuration with the **copy** *configuration_file* **running-configuration** command.

4. Copy the running configuration to the start-up configuration with the **copy running-config startup-config** command.

Additional References for Configuration Files

This section includes additional information related to managing configuration files.

Related Documents for Configuration Files

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	

CHAPTER 13

Scripting with Tcl

This chapter describes how to run tcl interactively and in scripts on a Cisco NX-OS device.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Guidelines and Limitations

Tcl has the following configuration guidelines and limitations:

Tclsh Command Help

Command help is not available for tcl commands. You can still access the help functions of Cisco NX-OS commands from within an interactive tcl shell.

This example shows the lack of tcl command help in an interactive tcl shell:

```
switch# tclsh
switch-tcl# set x 1
switch-tcl# puts ?
          ^
% Invalid command at '^' marker.
switch-tcl# configure ?
  <CR>
  session    Configure the system in a session
  terminal   Configure the system from terminal input

switch-tcl#
```

Note In the above example, the Cisco NX-OS command help function is still available but the tcl **puts** command returns an error from the help function.

Tclsh Command History

You can use the arrow keys on your terminal to access commands you previously entered in the interactive tcl shell.

Note The **tclsh** command history is not saved when you exit the interactive tcl shell.

Tclsh Tab Completion

You can use tab completion for Cisco NX-OS commands when you are running an interactive tcl shell. Tab completion is not available for tcl commands.

Tclsh CLI Command

Although you can directly access Cisco NX-OS commands from within an interactive tcl shell, you can only execute Cisco NX-OS commands in a tcl script if they are prepended with the tcl **cli** command.

In an interactive tcl shell, the following commands are identical and will execute properly:

```
switch-tcl# cli show module 1 | incl Mod
switch-tcl# cli "show module 1 | incl Mod"
switch-tcl# show module 1 | incl Mod
```

In a tcl script, you must prepend Cisco NX-OS commands with the tcl **cli** command as shown in this example:

```
set x 1
cli show module $x | incl Mod
cli "show module $x | incl Mod"
```

If you use the following commands in your script, the script will fail and the tcl shell will display an error:

```
show module $x | incl Mod
"show module $x | incl Mod"
```

Tclsh Command Separation

The semicolon (`;`) is the command separator in both Cisco NX-OS and tcl. To execute multiple Cisco NX-OS commands in a tcl command, you must enclose the Cisco NX-OS commands in quotes (`""`).

In an interactive tcl shell, the following commands are identical and will execute properly:

```
switch-tcl# cli "configure terminal ; interface loopback 10 ; description loop10"
switch-tcl# cli configure terminal ; cli interface loopback 10 ; cli description loop10
switch-tcl# cli configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.

switch(config-tcl)# cli interface loopback 10
switch(config-if-tcl)# cli description loop10
switch(config-if-tcl)#
```

In an interactive tcl shell, you can also execute Cisco NX-OS commands directly without prepending the tcl **cli** command:

```
switch-tcl# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(config-tcl)# interface loopback 10
switch(config-if-tcl)# description loop10
switch(config-if-tcl)#
```

Tcl Variables

You can use tcl variables as arguments to the Cisco NX-OS commands. You can also pass arguments into tcl scripts. Tcl variables are not persistent.

This example shows how to use a tcl variable as an argument to a Cisco NX-OS command:

```
switch# tclsh
switch-tcl# set x loop10
switch-tcl# cli "configure terminal ; interface loopback 10 ; description $x"
switch(config-if-tcl)#
```

Tclquit

The **tclquit** command exits the tcl shell regardless of which Cisco NX-OS command mode is currently active. You can also press **Ctrl-C** to exit the tcl shell. The **exit** and **end** commands change Cisco NX-OS command modes. The **exit** command will terminate the tcl shell only from the EXEC command mode.

Tclsh Security

The tcl shell is executed in a sandbox to prevent unauthorized access to certain parts of the Cisco NX-OS system. The system monitors CPU, memory, and file system resources being used by the tcl shell to detect events such as infinite loops, excessive memory utilization, and so on.

You configure the intial tcl environment with the **scripting tcl init** *init-file* command.

You can define the looping limits for the tcl environment with the **scripting tcl recursion-limit** *iterations* command. The default recursion limit is 1000 interations.

Information about Tcl

Tool Command Language (Tcl) is a scripting language created by John Ousterhout at the University of California, Berkeley. Tcl 8.5 was added to Cisco NX-OS Release 5.1(1) to provide scripting abilities. With tcl, you gain more flexibility in your use of the CLI commands on the device. You can use tcl to extract certain values in the output of a **show** command, perform switch configurations, run Cisco NX-OS commands in a loop, or define EEM policies in a script.

This section describes how to run tcl scripts or run tcl interactively on Cisco NX-OS devices.

Running the tclsh Command

You can run tcl commands from either a script or on the command line using the **tclsh** command.

Note You cannot create a tcl script file at the CLI prompt. You can create the script file on a remote device and copy it to the bootflash: directory on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**tclsh** [**bootflash:***filename* [*argument* ...]]
	Example:
	`switch# tclsh ?`
	`  <CR>`
	`  bootflash:  The file to run`
		Starts a tcl shell.
		If you run the **tclsh** command with no arguments, the shell runs interactively, reading tcl commands from standard input and printing command results and error messages to the standard output. You exit from the interactive tcl shell by entering **tclquit** or pressing **Ctrl-C**.
		If you enter the **tclsh** command with arguments, the first argument is the name of a script file that contains tcl commands and any additional arguments are made available to the script as variables.

Example

This example shows an interactive tcl shell:

```
switch# tclsh
switch-tcl# set x 1
switch-tcl# cli show module $x | incl Mod
Mod  Ports  Module-Type                         Model              Status
1    32     1/10 Gbps Ethernet Module           N7K-F132XP-15      ok
Mod  Sw              Hw
Mod  MAC-Address(es)                            Serial-Num
Mod  Online Diag Status
Left ejector CLOSE, Right ejector CLOSE, Module HW does support ejector based shutdown.
switch-tcl# exit
switch#
```

This example shows how to run a tcl script:

```
switch# show file bootflash:showmodule.tcl
set x 1
while {$x < 19} {
cli show module $x | incl Mod
set x [expr {$x + 1}]
}

switch# tclsh bootflash:showmodule.tcl
Mod  Ports  Module-Type                      Model              Status
1    32     1/10 Gbps Ethernet Module        N7K-F132XP-15      ok
Mod  Sw             Hw
Mod  MAC-Address(es)                         Serial-Num
Mod  Online Diag Status
Left ejector CLOSE, Right ejector CLOSE, Module HW does support ejector based shutdown.
switch#
```

Navigating Cisco NX-OS Modes from the tclsh Command

You can change modes in Cisco NX-OS while you are running an interactive tcl shell.

Procedure

	Command or Action	Purpose

Step 1	**tclsh**
	Example:
	`switch# tclsh`
	`switch-tcl#`
		Starts an interactive tcl shell.

Step 2	**configure terminal**
	Example:
	`switch-tcl# configure terminal`
	`switch(config-tcl)#`
		Runs a Cisco NX-OS command in the tcl shell, changing modes.
		Note The tcl prompt changes to indicate the Cisco NX-OS command mode.

Step 3	**tclquit**
	Example:
	`switch-tcl# tclquit`
	`switch#`
		Terminates the tcl shell and returns to the starting mode.

Example

This example shows how to change Cisco NX-OS modes from an interactive tcl shell:

```
switch# tclsh
switch-tcl# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config-tcl)# interface loopback 10
switch(config-if-tcl)# ?
  description  Enter description of maximum 80 characters
  inherit      Inherit a port-profile
  ip           Configure IP features
  ipv6         Configure IPv6 features
  logging      Configure logging for interface
  no           Negate a command or set its defaults
  rate-limit   Set packet per second rate limit
  shutdown     Enable/disable an interface
  this         Shows info about current object (mode's instance)
  vrf          Configure VRF parameters
  end          Go to exec mode
  exit         Exit from command interpreter
  pop          Pop mode from stack or restore from name
  push         Push current mode to stack or save it under name
  where        Shows the cli context you are in

switch(config-if-tcl)# description loop10
switch(config-if-tcl)# tclquit
Exiting Tcl
switch#
```

Tcl References

The following titles are provided for your reference:

- Mark Harrison (ed), *Tcl/Tk Tools*, O'Reilly Media, ISBN 1-56592-218-2, 1997
- Mark Harrison and Michael McLennan, *Effective Tcl/Tk Programming*, Addison-Wesley, Reading, MA, USA, ISBN 0-201-63474-0, 1998
- John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, USA, ISBN 0-201-63337-X, 1994.
- Brent B. Welch, *Practical Programming in Tcl and Tk*, Prentice Hall, Upper Saddle River, NJ, USA, ISBN 0-13-038560-3, 2003.
- J Adrian Zimmer, *Tcl/Tk for Programmers*, IEEE Computer Society, distributed by John Wiley and Sons, ISBN 0-8186-8515-8, 1998.